EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	0	hysom-donald$.in.	US-PGPUB; USPAT; EPO	OR	ON	2006/09/27 09:46
S2	0	hysom-ronald$.in.	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 15:06
S3	0	coleman-duncan$.in.	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 15:07
S4	1	ncr adj coporation$.as.	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 15:07
S5	15	teradata$.as.	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 15:54
S6	264	national adj cash adj register$.as.	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 15:55
S7	0	S6 and (data adj warehous$3)	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 15:55
S8	0	S6 and (information near model$4)	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 15:55
S9	31	S6 and quality	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 15:56
S10	1	S9 and model$4	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 15:56
S11	0	S9 and maturity	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 15:56
S12	0	S6 and metric	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 16:54
S13	26690	(information or data) near model$4	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 16:54
S14	238	S13 and (quality and maturity)	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 16:55
S15	120	S14 and (metric and capabilit$3)	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 16:56

EAST Search History

S16	79	S15 and ("3D" or "3-D" or (three adj dimension$2))	US-PGPUB; USPAT; EPO	OR	ON	2006/09/26 16:56
S17	5	("5500800"|"5530861"|"5551880"|"5655086"|"5662478").PN.	US-PGPUB; USPAT; EPO	OR	ON	2006/09/27 10:00
S18	2697	703/1,2.ccls.	US-PGPUB; USPAT	OR	ON	2006/09/27 10:00
S19	19	S18 and (data near warehous$3)	US-PGPUB; USPAT	OR	ON	2006/09/27 10:01
S20	17	S19 and model$4	US-PGPUB; USPAT	OR	ON	2006/09/27 13:03
S21	0	S18 and (model$4 near (information or data) near quality)	US-PGPUB; USPAT	OR	ON	2006/09/27 13:04
S22	538	S18 and (model$4 near (information or data))	US-PGPUB; USPAT	OR	ON	2006/09/27 13:04
S23	205	S22 and quality	US-PGPUB; USPAT	OR	ON	2006/09/27 13:04
S24	109	S23 and link$3	US-PGPUB; USPAT	OR	ON	2006/09/27 13:05
S25	81	S24 and dimension$3	US-PGPUB; USPAT	OR	ON	2006/09/27 13:18
S26	4750	data near warehous$3	US-PGPUB; USPAT	OR	ON	2006/09/27 13:27
S27	0	S26 and (model$4 near (quality adj metric))	US-PGPUB; USPAT	OR	ON	2006/09/27 13:19
S28	99	S26 and (information near quality)	US-PGPUB; USPAT	OR	ON	2006/09/27 13:20
S29	84	S28 and model$4	US-PGPUB; USPAT	OR	ON	2006/09/27 13:19
S30	461	S26 and ((information near quality) or (data near quality))	US-PGPUB; USPAT	OR	ON	2006/09/27 13:20
S31	415	S30 and model$4	US-PGPUB; USPAT	OR	ON	2006/09/27 13:20
S32	222	S31 and ("3D" or "3-d" or (three adj dimension$2))	US-PGPUB; USPAT	OR	ON	2006/09/27 13:21
S33	39	S32 and (metric and maturity and capabilit$3)	US-PGPUB; USPAT	OR	ON	2006/09/27 13:21
S34	3168	S26 and model$4	US-PGPUB; USPAT	OR	ON	2006/09/27 13:27
S35	399	S34 and visualization	US-PGPUB; USPAT	OR	ON	2006/09/27 13:27
S36	162	S35 and ("3D" or "3-d" or (three adj dimension$2))	US-PGPUB; USPAT	OR	ON	2006/09/27 13:28

EAST Search History

S37	38	S36 and geometric$2	US-PGPUB; USPAT	OR	ON	2006/09/27 13:37
S38	7536	(information near quality)	US-PGPUB; USPAT	OR	ON	2006/09/27 13:38
S39	2549	S38 and (database or repository or (data adj warehouse))	US-PGPUB; USPAT	OR	ON	2006/09/27 13:38
S40	1750	S39 and (functionality or integrity or sufficiency or availability or usability or contextual or complexity or responsiveness or adaptability or robustness or affordability or accessibility)	US-PGPUB; USPAT	OR	ON	2006/09/27 13:40
S41	1115	S40 and (power or meaningfulness or prodcutivity or utilization or scalability or resiliency or serviceability or (value adj added))	US-PGPUB; USPAT	OR	ON	2006/09/27 13:41
S42	640	S41 and (break$through or (leading adj edge) or advanced or intermediate or foundation)	US-PGPUB; USPAT	OR	ON	2006/09/27 13:42
S43	459	S42 and model$4	US-PGPUB; USPAT	OR	ON	2006/09/27 13:42
S44	447	S43 and (line or link)	US-PGPUB; USPAT	OR	ON	2006/09/27 13:43
S45	244	S44 and (surface or face)	US-PGPUB; USPAT	OR	ON	2006/09/27 13:43
S46	102	S45 and interactiv$3	US-PGPUB; USPAT	OR	ON	2006/09/27 13:43
S47	94	S46 and dimension$2	US-PGPUB; USPAT	OR	ON	2006/09/27 13:44
S48	94	S47 and state	US-PGPUB; USPAT	OR	ON	2006/09/27 14:20
S49	4562	data adj warehous$3	US-PGPUB; USPAT	OR	ON	2006/09/27 14:21
S50	468	S49 and (star or constellation)	US-PGPUB; USPAT	OR	ON	2006/09/27 14:21
S51	68	S50 and visualization	US-PGPUB; USPAT	OR	ON	2006/09/27 14:21
S52	67	S51 and dimension$2	US-PGPUB; USPAT	OR	ON	2006/09/27 16:31
S53	2697	703/1,2.ccls.	US-PGPUB; USPAT	OR	ON	2006/09/27 16:31
S54	864	703/1.ccls.	US-PGPUB; USPAT	OR	ON	2006/09/28 09:05
S55	447	S54 and ("3D" or "3-d" or (three adj dimension$2))	US-PGPUB; USPAT	OR	ON	2006/09/28 09:06

EAST Search History

S56	379	S55 and model$4	US-PGPUB; USPAT	OR	ON	2006/09/28 09:06
S57	131	S56 and (map or mapping)	US-PGPUB; USPAT	OR	ON	2006/09/28 09:07
S58	111	S57 and (surface or cell)	US-PGPUB; USPAT	OR	ON	2006/09/28 09:07
S59	103	S58 and (line or connection or link)	US-PGPUB; USPAT	OR	ON	2006/09/28 09:08
S60	31	S59 and visualizat$3	US-PGPUB; USPAT	OR	ON	2006/09/28 09:12
S61	65	S54 and (geometric$3 adj model$4)	US-PGPUB; USPAT	OR	ON	2006/09/28 09:12
S62	60	S61 and ("3D" or "3-D" or (three adj dimension$2))	US-PGPUB; USPAT	OR	ON	2006/09/28 09:13
S63	23	S62 and (map or mapping)	US-PGPUB; USPAT	OR	ON	2006/09/28 09:13
S64	21	S63 and (surface or cell)	US-PGPUB; USPAT	OR	ON	2006/09/28 09:13
S65	20	S64 and (line or connection or link)	US-PGPUB; USPAT	OR	ON	2006/09/28 09:30
S66	381	S54 and (data near warehous$3 or database or (data adj mart))	US-PGPUB; USPAT	OR	ON	2006/09/28 09:31
S67	206	S66 and ("3D" or "3-d" or (three adj dimension$2))	US-PGPUB; USPAT	OR	ON	2006/09/28 09:31
S68	143	S67 and (surface or cell)	US-PGPUB; USPAT	OR	ON	2006/09/28 09:31
S69	135	S68 and (line or connection or link or edge)	US-PGPUB; USPAT	OR	ON	2006/09/28 09:32
S70	29	S69 and visualization	US-PGPUB; USPAT	OR	ON	2006/09/28 09:38
S71	4643	(data adj warehous$3 or database or (data adj mart)) and cube	US-PGPUB; USPAT	OR	ON	2006/09/28 09:38
S72	2787	S71 and ("3D" or "3-d" or (three adj dimension$2))	US-PGPUB; USPAT	OR	ON	2006/09/28 09:39
S73	2376	S72 and (cell or surface)	US-PGPUB; USPAT	OR	ON	2006/09/28 09:39
S74	2081	S73 and (link or edge or connection)	US-PGPUB; USPAT	OR	ON	2006/09/28 09:39
S75	589	S74 and visualization	US-PGPUB; USPAT	OR	ON	2006/09/28 09:39
S76	475	S75 and (map or mapping)	US-PGPUB; USPAT	OR	ON	2006/09/28 09:40
S77	465148	(data adj warehous$3) or database	USPAT	OR	ON	2006/09/28 09:40
S78	1601	(data adj warehous$3)	USPAT	OR	ON	2006/09/28 09:40

EAST Search History

S79	270	S78 and cube	USPAT	OR	ON	2006/09/28 09:40
S80	139	S79 and ("3D" or "3-d" or (three adj dimension$2))	USPAT	OR	ON	2006/09/28 09:41
S81	90	S80 and (surface or cell)	USPAT	OR	ON	2006/09/28 09:41
S82	77	S81 and (link or edge or connection)	USPAT	OR	ON	2006/09/28 09:41
S83	75	S82 and (map$4 or assign$3)	USPAT	OR	ON	2006/09/28 09:42
S84	53	S83 and visualiz$5	USPAT	OR	ON	2006/09/28 09:42
S85	15	(US-20030144868-$ or US-20010039487-$).did. or (US-6714936-$ or US-6963826-$ or US-6995768-$ or US-6877006-$ or US-7003560-$ or US-7007029-$ or US-7089266-$ or US-4785399-$ or US-6629065-$ or US-6895371-$ or US-6581068-$ or US-6205447-$ or US-5918232-$).did.	US-PGPUB; USPAT	OR	ON	2006/09/28 11:07
S86	3	S85 and triangle	US-PGPUB; USPAT	OR	ON	2006/09/28 11:07

Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals. Paulraj Ponniah

ISBNs: 0-471-41254-6 (Hardback); 0-471-22162-7 (Electronic)

CHAPTER 15

OLAP IN THE DATA WAREHOUSE

CHAPTER OBJECTIVES

- Perceive the unqualified demand for online analytical processing (OLAP) and understand what drives this demand
- Review the major features and functions of OLAP in detail
- Grasp the intricacies of dimensional analysis and learn the meanings of hypercubes, drill-down and roll-up, and slice-and-dice
- Examine the different OLAP models and determine which model is suitable for your environment
- Consider OLAP implementation by studying the steps and the tools

In the earlier chapters we mentioned online analytical processing (OLAP) in passing. You had a glimpse of OLAP when we discussed the information delivery methods. You have some idea of what OLAP is and how it is used for complex analysis. As the name implies, OLAP has to do with the processing of data as it is manipulated for analysis. The data warehouse provides the best opportunity for analysis and OLAP is the vehicle for carrying out involved analysis. The data warehouse environment is also best for data access when analysis is carried out.

We now have the chance to explore OLAP in sufficient depth. In today's data warehousing environment, with such tremendous progress in analysis tools from various vendors, you cannot have a data warehouse without OLAP. It is unthinkable. Therefore, throughout this chapter, look out for the important topics.

First, you have to perceive what OLAP is and why it is absolutely essential. This will help you to better understand the features and functions of OLAP. We will discuss the major features and functions so that your grasp of OLAP may be firmed up. There are two major models for OLAP. You should know which model is most suitable for your computing and user environments. We will highlight the significance of each model, learn how to

implement OLAP in your data warehouse environment, investigate OLAP tools, and find out how to evaluate and get them for your users. Finally, we will discuss the implementation steps for OLAP.

DEMAND FOR ONLINE ANALYTICAL PROCESSING

Recall our discussions in Chapter 2 of the top-down and bottom-up approaches for building a data warehouse. In the top-down approach, you build the overall corporate-wide data repository using the entity-relationship (E-R) modeling technique. This enterprise-wide data warehouse feeds the departmental data marts that are designed using the dimensional modeling technique. In the bottom-up approach, you build several data marts using the dimensional modeling technique and the collection of these data marts forms the data warehouse environment for your company. Each of these two approaches has its advantages and shortcomings.

You also learned about a practical approach to building a conglomeration of supermarts with conformed and standardized data content. While adopting this approach, first you plan and define the requirements at the corporate level, build the infrastructure for the complete warehouse, and then implement one supermart at a time in a priority sequence. The supermarts are designed using the dimensional modeling technique.

As we have seen, a data warehouse is meant for performing substantial analysis using the available data. The analysis leads to strategic decisions that are the major reasons for building data warehouses in the first place. For performing meaningful analysis, data must be cast in a way suitable for analysis of the values of key indicators over time along business dimensions. Data structures designed using the dimensional modeling technique support such analysis.

In all the three approaches referred to above, the data marts rest on the dimensional model. Therefore, these data marts must be able to support dimensional analysis. In practice, these data marts seem to be adequate for basic analysis. However, in today's business conditions, we find that users need to go beyond such basic analysis. They must have the capability to perform far more complex analysis in less time. Let us examine how the traditional methods of analysis provided in a data warehouse are not sufficient and perceive what exactly is demanded by the users to stay competitive and to expand.

Need for Multidimensional Analysis

Let us quickly review the business model of a large retail operation. If you just look at daily sales, you soon realize that the sales are interrelated to many business dimensions. The daily sales are meaningful only when they are related to the dates of the sales, the products, the distribution channels, the stores, the sales territories, the promotions, and a few more dimensions. Multidimensional views are inherently representative of any business model. Very few models are limited to three dimensions or less. For planning and making strategic decisions, managers and executives probe into business data through scenarios. For example, they compare actual sales against targets and against sales in prior periods. They examine the breakdown of sales by product, by store, by sales territory, by promotion, and so on.

Decision makers are no longer satisfied with one-dimensional queries such as "How many units of Product A did we sell in the store in Edison, New Jersey?" Consider the fol-

lowing more useful query: How much revenue did the new Product X generate during the last three months, broken down by individual months, in the South Central territory, by individual stores, broken down by promotions, compared to estimates, and compared to the previous version of the product? The analysis does not stop with this single multidimensional query. The user continues to ask for further comparisons to similar products, comparisons among territories, and views of the results by rotating the presentation between columns and rows.

For effective analysis, your users must have easy methods of performing complex analysis along several business dimensions. They need an environment that presents a multidimensional view of data, providing the foundation for analytical processing through easy and flexible access to information. Decision makers must be able to analyze data along any number of dimensions, at any level of aggregation, with the capability of viewing results in a variety of ways. They must have the ability to drill down and roll up along the hierarchies of every dimension. Without a solid system for true multidimensional analysis, your data warehouse is incomplete.

In any analytical system, time is a critical dimension. Hardly any query is executed without having time as one of the dimensions along which analysis is performed. Further, time is a unique dimension because of its sequential nature—November always comes after October. Users monitor performance over time, as for example, performance this month compared to last month, or performance this month compared with performance the same month last year.

Another point about the uniqueness of the time dimension is the way in which the hierarchies of the dimension work. A user may look for sales in March and may also look for sales for the first four months of the year. In the second query for sales for the first four months, the implied hierarchy at the next higher level is an aggregation taking into account the sequential nature of time. No user looks for sales of the first four stores or the last three stores. There is no implied sequence in the store dimension. True analytical systems must recognize the sequential nature of time.

Fast Access and Powerful Calculations

Whether a user's request is for monthly sales of all products along all geographical regions or for year-to-date sales in a region for a single product, the query and analysis system must have consistent response times. Users must not be penalized for the complexity of their analysis. Both the size of the effort to formulate a query or the amount of time to receive the result sets must be consistent irrespective of the query types.

Let us take an example to understand how speed of the analysis process matters to users. Imagine a business analyst looking for reasons why profitability dipped sharply in the recent months in the entire enterprise. The analyst starts this analysis by querying for the overall sales for the last five months for the entire company, broken down by individual months. The analyst notices that although the sales do not show a drop, there is a sharp reduction in profitability for the last three months. The analysis proceeds further when the analyst wants to find out which countries show reductions. The analyst requests a breakdown of sales by major worldwide regions and notes that the European region is responsible for the reduction in profitability. Now the analyst senses that clues are becoming more pronounced and looks for a breakdown of the European sales by individual countries. The analyst finds that the profitability has increased for a few countries, decreased sharply for some other countries, and been stable for the rest. At this point, the analyst introduces an-

other dimension into the analysis. Now the analyst wants the breakdown of profitability for the European countries by country, month, and product. This step brings the analyst closer to the reason for the decline in the profitability. The analyst observes that the countries in the European Union (EU) show very sharp declines in profitability for the last two months. Further queries reveal that manufacturing and other direct costs remain at the usual levels but the indirect costs have shot up. The analyst is now able to determine that the decline is due to the additional tax levies on some products in the EU. The analyst has also determined the exact effect of the levies so far. Strategic decisions follow on how to deal with the decline in profitability.

Now please look at Figure 15-1 showing the steps through the single analysis session. How many steps are there? Many steps, but a single analysis session and train of thought. Each step in this train of thought constitutes a query. The analyst formulates each query, executes it, waits for the result set to appear on the screen, and studies the result set. Each query is interactive because the result set from one query forms the basis for the next query. In this manner of querying, the user cannot maintain the train of thought unless the momentum is preserved. Fast access is absolutely essential for an effective analytical processing environment.

Did you notice that none of the queries in the above analysis session included any serious calculations? This is not typical. In a real-world analysis session, many of the queries require calculations, sometimes complex calculations. What is the implication here? An effective analytical processing environment must not only be fast and flexible, but it must also support complex and powerful calculations.

What follows is a list of typical calculations that get included in the query requests:

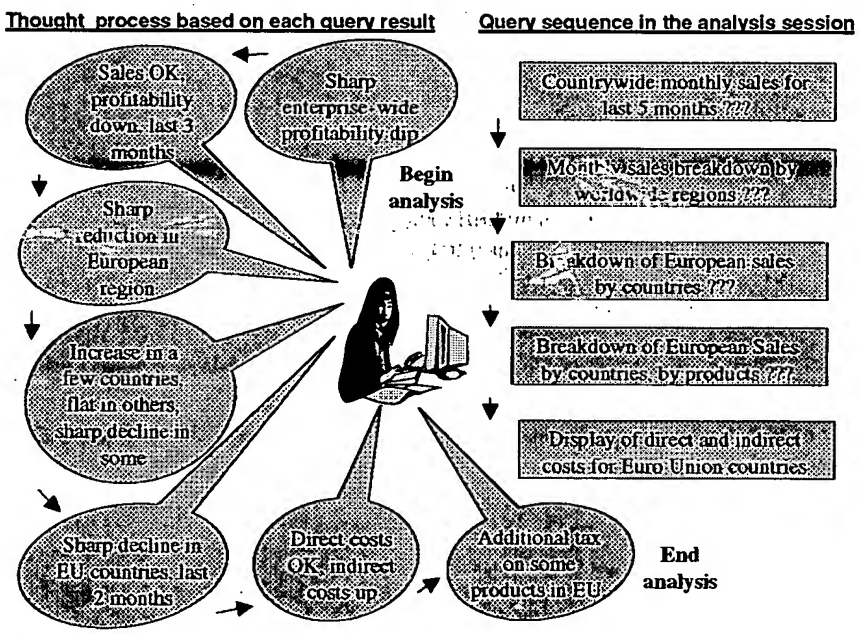


Figure 15-1 Query steps in an analysis session.

- Roll-ups to provide summaries and aggregations along the hierarchies of the dimensions.
- Drill-downs from the top level to the lowest along the hierarchies of the dimensions, in combinations among the dimensions.
- Simple calculations, such as computation of margins (sales minus costs).
- Share calculations to compute the percentage of parts to the whole.
- Algebraic equations involving key performance indicators.
- Moving averages and growth percentages.
- Trend analysis using statistical methods.

Limitations of Other Analysis Methods

You now have a fairly good grip on the types of requirements of users to execute queries and perform analysis. First and foremost, the information delivery system must be able to present multidimensional views of the data. Then the information delivery system must enable the users to use the data by analyzing it along multiple dimensions and their hierarchies in a myriad of ways. And this facility must be fast. It must be possible for the users to perform complex calculations.

Let us understand why the traditional tools and methods are not up to the task when it comes to complex analysis and calculations. What information methods are we familiar with? Of course, the earliest method was the medium of reports. Then came spreadsheets with all their functionality and features. SQL has been the accepted interface for retrieving and manipulating data from relational databases. These methods are used in OLTP systems and in data warehouse environments. Now, when we discuss multidimensional analysis and complex calculations, how suitable are these traditional methods?

First, let us look at the characteristics of the OLTP and data warehouse environments. When we mention the data warehouse environment here, we are not referring to heavy multidimensional analysis and complex calculations. We are only referring to the environment with simple queries and routine reports. Please see Figure 15-2 showing the characteristics of the OLTP and the basic data warehouse environments as they relate to information delivery needs.

Now consider information retrieval and manipulation in these two environments. What are the standard methods of information delivery? Reports, spreadsheets, and online displays. What is the standard data access interface? SQL. Let us review these and determine if they are adequate for multidimensional analysis and complex calculations.

Report writers provide two key functions: the ability to point and click for generating and issuing SQL calls, and the capability to format the output reports. However, report writers do not support multidimensionality. With basic report writers, you cannot drill down to lower levels in the dimensions. That will have to come from additional reports. You cannot rotate the results by switching rows and columns. The report writers do not provide aggregate navigation. Once the report is formatted and run, you cannot alter the presentation of the result data sets.

If report writers are not the tools or methods we are looking for, how about spreadsheets for calculations and the other features needed for analysis? Spreadsheets, when they first appeared, were positioned as analysis tools. You can perform "what if" analysis with spreadsheets. When you modify the values in some cells, the values in other related cells automatically change. What about aggregations and calculations? Spreadsheets with

CHARACTERISTICS	OLTP SYSTEMS	DATA WAREHOUSE
Analytical capabilities	Very low	Moderate
Data for a single session	Very limited	Small to medium size
Size of result set	Small	Large
Response time	Very fast	Fast to moderate
Data granularity	Detail	Detail and summary
Data currency	Current	Current and historical
Access method	Predefined	Predefined and ad hoc
Basic motivation	Collect and input data	Provide information
Data model	Design for data updates	Design for queries
Optimization of database	For transactions	For analysis
Update frequency	Very frequent	Generally read-only
Scope of user interaction	Single transactions	Throughout data content

Figure 15-2 OLTP and data warehouse environments.

their add-in tools can perform some forms of aggregations and also do a variety of calculations. Third party tools have also enhanced spreadsheet products to present data in three-dimensional formats. You can view rows, columns, and pages on spreadsheets. For example, the rows can represent products, the columns represent stores, and the pages represent the time dimension in months. Modern spreadsheet tools offer pivot tables or n-way cross-tabs.

Even with enhanced functionality using add-ins, spreadsheets are still very cumbersome to use. Take an analysis involving the four dimensions of store, product, promotion, and time. Let us say each dimension contains an average of five hierarchical levels. Now try to build an analysis to retrieve data and present it as spreadsheets showing all the aggregation levels and multidimensional views, and also using even simple calculations. You can very well imagine how much effort it would take for this exercise. Now what if your user wants to change the navigation and do different roll-ups and drill-downs. The limitations of spreadsheets for multidimensional analysis and complex calculations are quite evident.

Let us now turn our attention to SQL (Structured Query Language). Although it might have been the original goal of SQL to be the end-user query language, now everyone agrees that the language is too abstruse even for sophisticated users. Third-party products attempt to extend the capabilities of SQL and hide the syntax from the users. Users can formulate their queries through GUI point-and-click methods or by using natural language syntax. Nevertheless, SQL vocabulary is ill-suited for analyzing data and exploring relationships. Even basic comparisons prove to be difficult in SQL.

Meaningful analysis such as market exploration and financial forecasting typically involve retrieval of large quantities of data, performing calculations, and summarizing the data on the fly. Perhaps, even the detailed analysis may be achieved by using SQL for re-

trieval and spreadsheets for presenting the results. But here is the catch: in a real-world analysis session, many queries follow one after the other. Each query may translate into a number of intricate SQL statements, with each of the statements likely to invoke full table scans, multiple joins, aggregations, groupings, and sorting. Analysis of the type we are discussing requires complex calculations and handling time series data. SQL is notably weak in these areas. Even if you can imagine an analyst accurately formulating such complex SQL statements, the overhead on the systems would still be enormous and seriously impact the response times.

OLAP is the Answer

Users certainly need the ability to perform multidimensional analysis with complex calculations, but we find that the traditional tools of report writers, query products, spreadsheets, and language interfaces are distressfully inadequate. What is the answer? Clearly, the tools being used in the OLTP and basic data warehouse environments do not match up to the task. We need different set of tools and products that are specifically meant for serious analysis. We need OLAP in the data warehouse.

In this chapter, we will thoroughly examine the various aspects of OLAP. We will come up with formal definitions and detailed characteristics. We will highlight all the features and functions. We will explore the different OLAP models. But now that you have an initial appreciation for OLAP, let us list the basic virtues of OLAP to justify our proposition.

- Enables analysts, executives, and managers to gain useful insights from the presentation of data.
- Can reorganize metrics along several dimensions and allow data to be viewed from different perspectives.
- Supports multidimensional analysis.
- Is able to drill down or roll up within each dimension.
- Is capable of applying mathematical formulas and calculations to measures.
- Provides fast response, facilitating speed-of-thought analysis.
- Complements the use of other information delivery techniques such as data mining.
- Improves the comprehension of result sets through visual presentations using graphs and charts.
- Can be implemented on the Web.
- Designed for highly interactive analysis.

Even at this stage, you will further appreciate the nature and strength of OLAP by studying a typical OLAP session (see Figure 15-3). The analyst starts with a query requesting a high-level summary by product line. Next, the user moves to drilling down for details by year. In the following step, the analyst pivots the data to view totals by year rather than totals by product line. Even in such a simple example, you observe the power and features of OLAP.

OLAP Definitions and Rules

Where did the term OLAP originate? We know that multidimensionality is at the core of OLAP systems. We have also mentioned some other basic features of OLAP. Is it a vague

LINE	TOTAL SALES
Clothing	$12,836,450
Electronics	$16,068,300
Video	$21,262,190
Kitchen	$17,704,400
Appliances	$19,600,800
Total	$87,472,140

1 High level summary by product line

2 Drill down by year

LINE	1998	1999	2000	TOTAL
Clothing	$3,457,000	$3,590,050	$5,789,400	$12,836,450
Electronics	$5,894,800	$4,078,900	$6,094,600	$16,068,300
Video	$7,198,700	$6,057,890	$8,005,600	$21,262,190
Kitchen	$4,875,400	$5,894,500	$6,934,500	$17,704,400
Appliances	$5,947,300	$6,104,500	$7,549,000	$19,600,800
Total	$27,373,200	$25,725,840	$34,373,100	$87,472,140

3 Rotate columns to rows

YEAR	Clothing	Electronics	Video	Kitchen	Appliances	TOTAL
1998	$3,457,000	$5,894,800	$7,198,700	$4,875,400	$5,947,300	$27,373,200
1999	$3,590,050	$4,078,900	$6,057,890	$5,894,500	$6,104,500	$25,725,840
2000	$5,789,400	$6,094,600	$8,005,600	$6,934,500	$7,549,000	$34,373,100
Total	$12,836,450	$16,068,300	$21,262,190	$17,704,400	$19,600,800	$87,472,140

Figure 15-3 Simple OLAP session.

collection of complex factors for serious analysis? Is there a formal definition and a set of fundamental guidelines identifying OLAP systems?

The term OLAP or online analytical processing was introduced in a paper entitled "Providing On-Line Analytical Processing to User Analysts," by Dr. E. F. Codd, the acknowledged "father" of the relational database model. The paper, published in 1993, defined 12 rules or guidelines for an OLAP system. Later, in 1995, six additional rules were included. We will discuss these rules. Before that, let us look for a short and precise definition for OLAP. Such a succinct definition comes from the OLAP council, which provides membership, sponsors research, and promotes the use of OLAP. Here is the definition:

> On-Line Analytical Processing (OLAP) is a category of software technology that enables analysts, managers and executives to gain insight into data through fast, consistent, interactive access in a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user.

The definition from the OLAP council contains all the key ingredients. Speed, consistency, interactive access, and multiple dimensional views—all of these are principal elements. As one trade magazine described it in 1995, OLAP is a fancy term for multidimensional analysis.

The guidelines proposed by Dr. Codd form the yardstick for measuring any sets of OLAP tools and products. A true OLAP system must conform to these guidelines. When

your project team is looking for OLAP tools, it can prioritize these guidelines and select tools that meet the set of criteria at the top of your priority list. First, let us consider the initial twelve guidelines for an OLAP system:

Multidimensional Conceptual View. Provide a multidimensional data model that is intuitively analytical and easy to use. Business users' view of an enterprise is multidimensional in nature. Therefore, a multidimensional data model conforms to how the users perceive business problems.

Transparency. Make the technology, underlying data repository, computing architecture, and the diverse nature of source data totally transparent to users. Such transparency, supporting a true open system approach, helps to enhance the efficiency and productivity of the users through front-end tools that are familiar to them.

Accessibility. Provide access only to the data that is actually needed to perform the specific analysis, presenting a single, coherent, and consistent view to the users. The OLAP system must map its own logical schema to the heterogeneous physical data stores and perform any necessary transformations.

Consistent Reporting Performance. Ensure that the users do not experience any significant degradation in reporting performance as the number of dimensions or the size of the database increases. Users must perceive consistent run time, response time, or machine utilization every time a given query is run.

Client/Server Architecture. Conform the system to the principles of client/server architecture for optimum performance, flexibility, adaptability, and interoperability. Make the server component sufficiently intelligent to enable various clients to be attached with a minimum of effort and integration programming.

Generic Dimensionality. Ensure that every data dimension is equivalent in both structure and operational capabilities. Have one logical structure for all dimensions. The basic data structure or the access techniques must not be biased toward any single data dimension.

Dynamic Sparse Matrix Handling. Adapt the physical schema to the specific analytical model being created and loaded that optimizes sparse matrix handling. When encountering a sparse matrix, the system must be able to dynamically deduce the distribution of the data and adjust the storage and access to achieve and maintain consistent level of performance.

Multiuser Support. Provide support for end users to work concurrently with either the same analytical model or to create different models from the same data. In short, provide concurrent data access, data integrity, and access security.

Unrestricted Cross-dimensional Operations. Provide ability for the system to recognize dimensional hierarchies and automatically perform roll-up and drill-down operations within a dimension or across dimensions. Have the interface language allow calculations and data manipulations across any number of data dimensions, without restricting any relations between data cells, regardless of the number of common data attributes each cell contains.

Intuitive Data Manipulation. Enable consolidation path reorientation (pivoting), drill-down and roll-up, and other manipulations to be accomplished intuitively and directly via point-and-click and drag-and-drop actions on the cells of the analytical model. Avoid the use of a menu or multiple trips to a user interface.

Flexible Reporting. Provide capabilities to the business user to arrange columns, rows, and cells in a manner that facilitates easy manipulation, analysis, and synthesis of information. Every dimension, including any subsets, must be able to be displayed with equal ease.

Unlimited Dimensions and Aggregation Levels. Accommodate at least fifteen, preferably twenty, data dimensions within a common analytical model. Each of these generic dimensions must allow a practically unlimited number of user-defined aggregation levels within any given consolidation path.

In addition to these twelve basic guidelines, also take into account the following requirements, not all distinctly specified by Dr. Codd.

Drill-through to Detail Level. Allow a smooth transition from the multidimensional, preaggregated database to the detail record level of the source data warehouse repository.

OLAP Analysis Models. Support Dr. Codd's four analysis models: exegetical (or descriptive), categorical (or explanatory), contemplative, and formulaic.

Treatment of Nonnormalized Data. Prohibit calculations made within an OLAP system from affecting the external data serving as the source.

Storing OLAP Results. Do not deploy write-capable OLAP tools on top of transactional systems.

Missing Values. Ignore missing values, irrespective of their source.

Incremental Database Refresh. Provide for incremental refreshes of the extracted and aggregated OLAP data.

SQL Interface. Seamlessly integrate the OLAP system into the existing enterprise environment.

OLAP Characteristics

Let us summarize in simple terms what we have covered so far. We explored why the business users absolutely need online analytical processing. We examined why the other methods of information delivery do not satisfy the requirements for multidimensional analysis with powerful calculations and fast access. We discussed how OLAP is the answer to satisfy these requirements. We reviewed the definitions and authoritative guidelines for the OLAP system.

Before we get into a more detailed discussion of the major features of OLAP systems, let us list the most fundamental characteristics in plain language. OLAP systems

- let business users have a multidimensional and logical view of the data in the data warehouse,
- facilitate interactive query and complex analysis for the users,
- allow users to drill down for greater details or roll up for aggregations of metrics along a single business dimension or across multiple dimensions,
- provide ability to perform intricate calculations and comparisons, and
- present results in a number of meaningful ways, including charts and graphs.

MAJOR FEATURES AND FUNCTIONS

Very often, you are faced with the question of whether OLAP is not just data warehousing in a nice wrapper? Can you not consider online analytical processing as just an information delivery technique and nothing more? Is it not another layer in the data warehouse, providing interface between the data and the users? In some sense, OLAP is an information delivery system for the data warehouse. But OLAP is much more than that. A data warehouse stores data and provides simpler access to the data. An OLAP system complements the data warehouse by lifting the information delivery capabilities to new heights.

General Features

In this section, we will pay special attention to a few major features and functions of OLAP systems. You will gain greater insight into dimensional analysis, find deeper meanings about the necessity for drill-downs and roll-ups during analysis sessions and gain greater appreciation for the role of slicing and dicing operations in analysis. Before getting into greater details about these, let us recapitulate the general features of OLAP. Please go to Figure 15-4 and note the summary. Also note the distinction between basic features and advanced features. The list shown in the figure includes the general features you observe in practice in most OLAP environments. Please use the list as a quick checklist of features your project team must consider for your OLAP system.

Dimensional Analysis

By this time, you are perhaps tired of the term "dimensional analysis." We had to use the term a few times so far. You have been told that dimensional analysis is a strong suit in the

BASIC FEATURES	Multidimensional analysis	Consistent performance	Fast response times for interactive queries
	Drill-down and roll-up	Navigation in and out of details	Slice-and-dice or rotation
	Multiple view modes	Easy scalability	Time intelligence (year-to-date, fiscal period)
ADVANCED FEATURES	Powerful calculations	Cross-dimensional calculations	Pre-calculation or pre-consolidation
	Drill-through across dimensions or details	Sophisticated presentation & displays	Collaborative decision making
	Derived data values through formulas	Application of alert technology	Report generation with agent technology

Figure 15-4 General features of OLAP.

arsenal of OLAP. Any OLAP system devoid of multidimensional analysis is utterly useless. So try to get a clear picture of the facility provided in OLAP systems for dimensional analysis.

Let us begin with a simple STAR schema. This STAR schema has three business dimensions, namely, product, time, and store. The fact table contains sales. Please see Figure 15-5 showing the schema and a three-dimensional representation of the model as a cube, with products on the X-axis, time on the Y-axis, and stores on the Z-axis. What are the values represented along each axis? For example, in the STAR schema, time is one of the dimensions and month is one of the attributes of the time dimension. Values of this attribute month are represented on the Y-axis. Similarly, values of the attributes product name and store name are represented on the other two axes.

This schema with just three business dimensions does not even look like a star. Nevertheless, it is a dimensional model. From the attributes of the dimension tables, pick the attribute product name from the product dimension, month from the time dimension, and store name from the store dimension. Now look at the cube representing the values of these attributes along the primary edges of the physical cube. Go further and visualize the sales for coats in the month of January at the New York store to be at the intersection of the three lines representing the product: coats, month: January, and store: New York.

If you are displaying the data for sales along these three dimensions on a spreadsheet, the columns may display the product names, the rows the months, and pages the data along the third dimension of store names. See Figure 15-6 showing a screen display of a page of this three-dimensional data.

The page displayed on the screen shows a slice of the cube. Now look at the cube and move along a slice or plane passing through the point on the Z-axis representing store: New York. The intersection points on this slice or plane relate to sales along product and

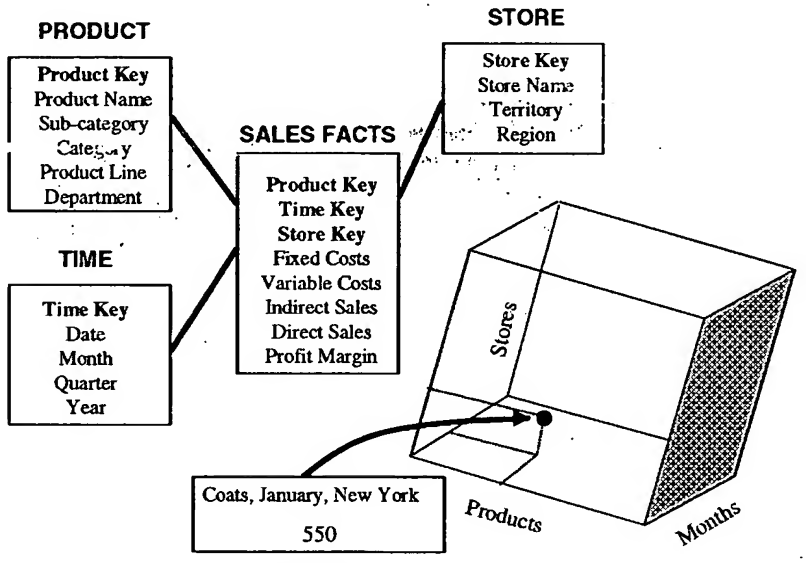


Figure 15-5 Simple STAR schema.

Store: New York

PAGES: STORE dimension

Products

COLUMNS: PRODUCT dimension

ROWS: TIME dimension

Months

	Hats	Coats	Jackets	Dresses	Shirts	Slacks
Jan	200	550	350	500	520	490
Feb	210	480	390	510	530	500
Mar	190	480	380	480	500	470
Apr	190	430	350	490	510	480
May	160	530	320	530	550	520
Jun	150	450	310	540	560	330
Jul	130	480	270	550	570	250
Aug	140	570	250	650	670	230
Sep	160	470	240	630	650	210
Oct	170	480	260	610	630	250
Nov	180	520	280	680	700	260
Dec	200	560	320	750	770	310

Figure 15-6 A Three-dimensional display.

time business dimensions for store: New York. Try to relate these sale numbers to the slice on the cube representing store: New York.

Now we have a way of depicting three business dimensions and a single fact on a two-dimensional page and also on a three-dimensional cube. The numbers in each cell on the page are the sale numbers. What could be the types of multidimensional analysis on this particular set of data? What types of queries could be run during the course of analysis sessions? You could get sale numbers along the hierarchies of a combination of the three business dimensions of product, store, and time. You could perform various types of three-dimensional analysis of sales. The results of queries during analysis sessions will be displayed on the screen with the three dimensions represented in columns, rows, and pages. The following is a sample of simple queries and the result sets during a multidimensional analysis session.

Query

Display the total sales of all products for past five years in all stores.

Display of Results

Rows: Year numbers 2000, 1999, 1998, 1997, 1996

Columns: Total Sales for all products

Page: One store per page

Query

Compare total sales for all stores, product by product, between years 2000 and 1999.

Display of Results

Rows: Year numbers 2000, 1999; difference; percentage increase or decrease

Columns: One column per product, showing all products

Page: All stores

Query

Show comparison of total sales for all stores, product by product, between years 2000 and 1999 only for those products with reduced sales.

Display of Results

Rows: Year numbers 2000, 1999; difference; percentage decrease

Columns: One column per product, showing only the qualifying products

Page: All stores

Query

Show comparison of sales by individual stores, product by product, between years 2000 and 1999 only for those products with reduced sales.

Display of Results

Rows: Year numbers 2000, 1999; difference; percentage decrease

Columns: One column per product, showing only the qualifying products

Page: One store per page

Query

Show the results of the previous query, but rotating and switching the columns with rows.

Display of Results

Rows: One row per product, showing only the qualifying products

Columns: Year numbers 2000, 1999; difference; percentage decrease

Page: One store per page

Query

Show the results of the previous query, but rotating and switching the pages with rows.

Display of Results

Rows: One row per store

Columns: Year numbers 2000, 1999; difference; percentage decrease

Page: One product per page, displaying only the qualifying products.

This multidimensional analysis can continue on until the analyst determines how many products showed reduced sales and which stores suffered the most.

In the above example, we had only three business dimensions and each of the dimensions could, therefore, be represented along the edges of a cube or the results displayed as columns, rows, and pages. Now add another business dimension, promotion. That will bring the number of business dimensions to four. When you have three business dimensions, you are able to represent these three as a cube with each edge of the cube denoting one dimension. You are also able to display the data on a spreadsheet with two dimensions as rows and columns and the third dimension as pages. But when you have four dimensions or more, how can you represent the data? Obviously, a three-dimensional cube does not work. And you also have a problem when trying to display the data on a spreadsheet as rows, columns, and pages. So what about multidimensional analysis when there are more than three dimensions? This leads us to a discussion of hypercubes.

What are Hypercubes?

Let us begin with the two business dimensions of product and time. Usually, business users wish to analyze not just sales but other metrics as well. Assume that the metrics to be analyzed are fixed cost, variable cost, indirect sales, direct sales, and profit margin. These are five common metrics.

The data described here may be displayed on a spreadsheet showing metrics as columns, time as rows, and products as pages. Please see Figure 15-7 showing a sample page of the spreadsheet display. In the figure, please also note the three straight lines, two of which represent the two business dimensions and the third, the metrics. You can independently move up or down along the straight lines. Some experts refer to this representation of a multidimension as a multidimensional domain structure (MDS).

The figure also shows a cube representing the data points along the edges. Relate the three straight lines to the three edges of the physical cube. Now the page you see in the figure is a slice passing through a single product and the divisions along the other two straight lines shown on the page as columns and rows. With three groups of data—two groups of business dimensions and one group of metrics—we can easily visualize the data as being along the three edges of a cube.

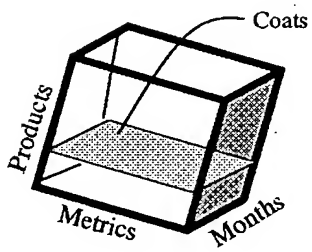
Now add another business dimension to the model. Let us add the store dimension. That results in three business dimensions plus the metrics data. How can you represent these four groups as edges of a three-dimensional cube? How do you represent a four-dimensional model with data points along the edges of a three-dimensional cube? How do you slice the data to display pages?

PRODUCT: Coats

PAGES: PRODUCT dimension COLUMNS: Metrics

ROWS: TIME dimension

	Fixed Cost	Variable Cost	Indirect Sales	Direct Sales	Profit Margin
Jan	340	110	230	320	100
Feb	270	90	200	260	100
Mar	310	100	210	270	70
Apr	340	110	210	320	80
May	330	110	230	300	90
Jun	260	90	150	300	100
Jul	310	100	180	300	70
Aug	300	130	210	360	60
Sep	300	100	180	290	70
Oct	310	100	170	310	70
Nov	330	110	210	310	80
Dec	350	120	200	360	90



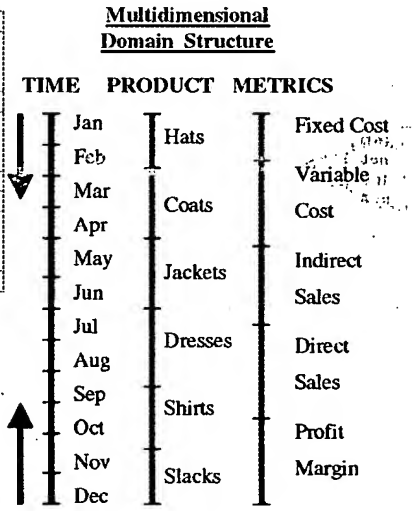


Figure 15-7 Display of columns, rows, and pages.

This is where an MDS diagram comes in handy. Now you need not try to perceive four-dimensional data as along the edges of the three-dimensional cube. All you have to do is draw four straight lines to represent the data as an MDS. These four lines represent the data. Please see Figure 15-8. By looking at this figure, you realize that the metaphor of a physical cube to represent data breaks down when you try to represent four dimensions. But, as you see, the MDS is well suited to represent four dimensions. Can you think of the four straight lines of the MDS intuitively to represent a "cube" with four primary edges? This intuitive representation is a hypercube, a representation that accommodates more than three dimensions. At a lower level of simplification, a hypercube can very well accommodate three dimensions. A hypercube is a general metaphor for representing multi-dimensional data.

You now have a way of representing four dimensions as a hypercube. The next question relates to display of four-dimensional data on the screen. How can you possibly show four dimensions with only three display groups of rows, columns, and pages? Please turn your attention to Figure 15-9. What do you notice about the display groups? How does the display resolve the problem of accommodating four dimensions with only three display groups? By combining multiple logical dimensions within the same display group. Notice how product and metrics are combined to display as columns. The displayed page represents the sales for store: New York.

Let us look at just one more example of an MDS representing a hypercube. Let us move up to six dimensions. Please study Figure 15-10 with six straight lines showing the data representations. The dimensions shown in this figure are product, time, store, promotion, customer demographics, and metrics.

There are several ways you can display six-dimensional data on the screen. Figure 15-11 illustrates one such six-dimensional display. Please study the figure carefully. Notice how product and metrics are combined and represented as columns, store and time are combined as rows, and demographics and promotion as pages.

We have reviewed two specific issues. First, we have noted a special method for repre-

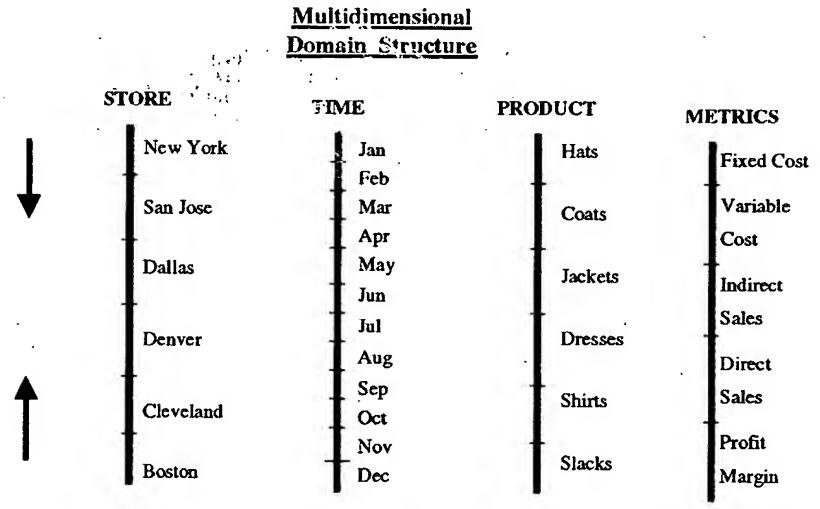
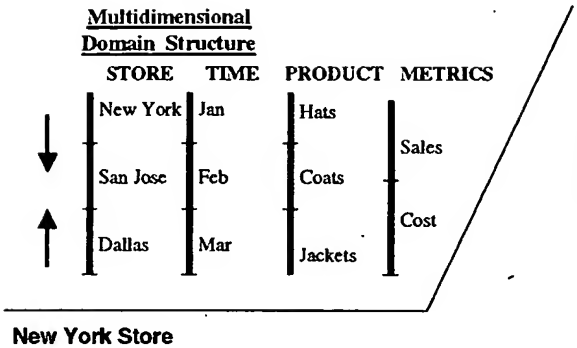


Figure 15-8 MDS for four dimensions.



New York Store

	Hats:Sales	Hats:Cost	Coats:Sales	Coats:Cost	Jackets:Sales	Jackets:Cost
Jan	450	350	550	450	500	400
Feb	380	280	460	360	400	320
Mar	400	310	480	410	450	400

Figure 15-9 Page displays for four-dimensional data.

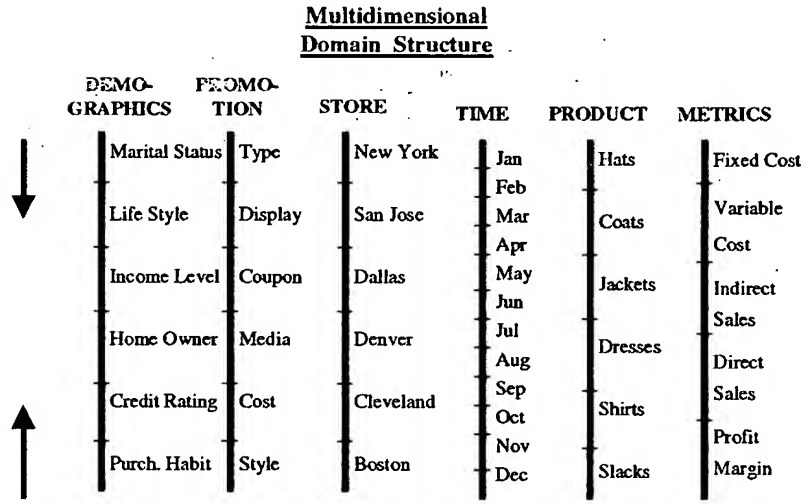
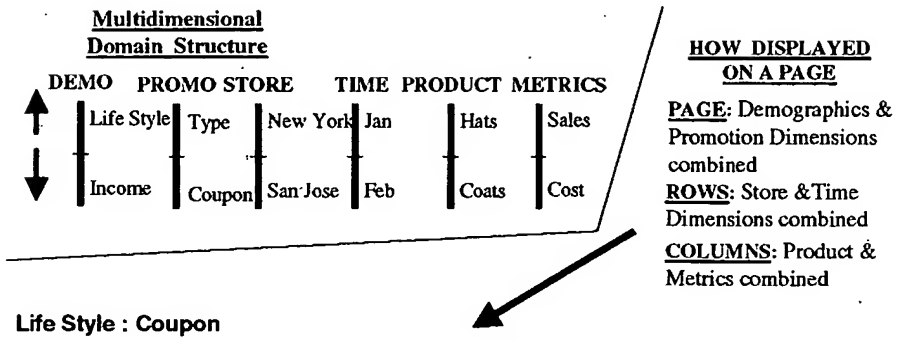


Figure 15-10 Six-dimensional MDS.



Life Style : Coupon

		Hats Sales	Hats Cost	Coats Sales	Coats Cost
New York	Jan	220	170	270	220
	Feb	190	140	230	180
Boston	Jan	200	160	240	200
	Feb	180	130	220	170

Figure 15-11 Page displays for six-dimensional data.

senting a data model with more than three dimensions using an MDS. This method is an intuitive way of showing a hypercube. A model with three dimensions can be represented by a physical cube. But a physical cube is limited to only three dimensions or less. Second, we have also discussed the methods for displaying the data on a flat screen when the number of dimensions is three or more. Building on the resolution of these two issues, let us now move on to two very significant aspects of multidimensional analysis. One of these is the drill-down and roll-up exercise; the other is the slice-and-dice operation.

Drill-Down and Roll-Up

Return to Figure 15-5. Look at the attributes of the product dimension table of the STAR schema. In particular, note these specific attributes of the product dimension: product name, subcategory, category, product line, and department. These attributes signify an ascending hierarchical sequence from product name to department. A department includes product lines, a product line includes categories, a category includes subcategories, and each subcategory consists of products with individual product names. In an OLAP system, these attributes are called hierarchies of the product dimension.

OLAP systems provide drill-down and roll-up capabilities. Try to understand what we mean by these capabilities with reference to above example. Please see Figure 15-12 illustrating these capabilities with reference to the product dimension hierarchies. Note the different types of information given in the figure. It shows the rolling up to higher hierarchical levels of aggregation and the drilling down to lower levels of detail. Also note the sales numbers shown alongside. These are sales for one particular store in one particular month at these levels of aggregation. The sale numbers you notice as you go down the hierarchy are for a single department, a single product line, a single category, and so on. You drill down to get the lower level breakdown of sales. The figure also shows the drill-across

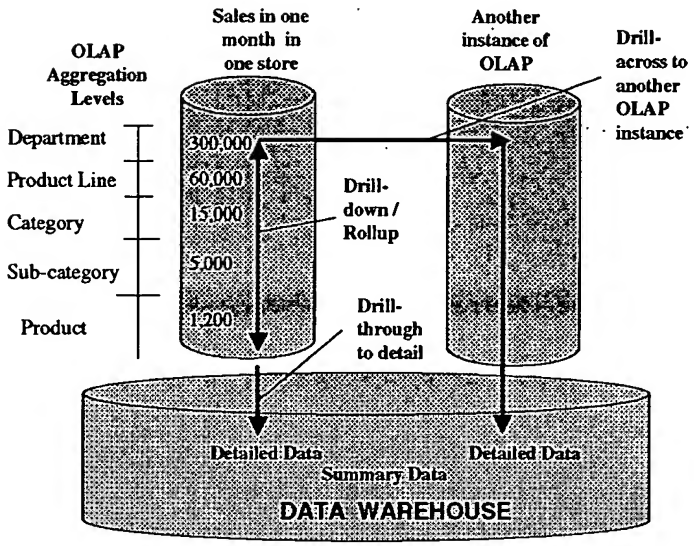


Figure 15-12 Roll-up and drill-down features of OLAP.

to another OLAP summarization using a different set of hierarchies of other dimensions. Notice also the drill-through to the lower levels of granularity, as stored in the source data warehouse repository. Roll-up, drill-down, drill-across, and drill-through are extremely useful features of OLAP systems supporting multidimensional analysis.

On more question remains. While you are rolling up or drilling down, how do the page displays change on the spreadsheets? For example, return to Figure 15-6 and look at the

Store: New York

Sub-categories

PAGES: STORE dimension

COLUMNS: PRODUCT dimension

ROWS: TIME dimension

Months

	Outer	Dress	Casual
Jan	1,100	1,020	490
Feb	1,080	1,040	500
Mar	1,050	980	470
Apr	970	1,000	480
May	1,010	1,080	520
Jun	910	1,100	330
Jul	880	1,120	250
Aug	960	1,320	230
Sep	870	1,280	210
Oct	910	1,240	250
Nov	980	1,380	260
Dec	1,080	1,520	310

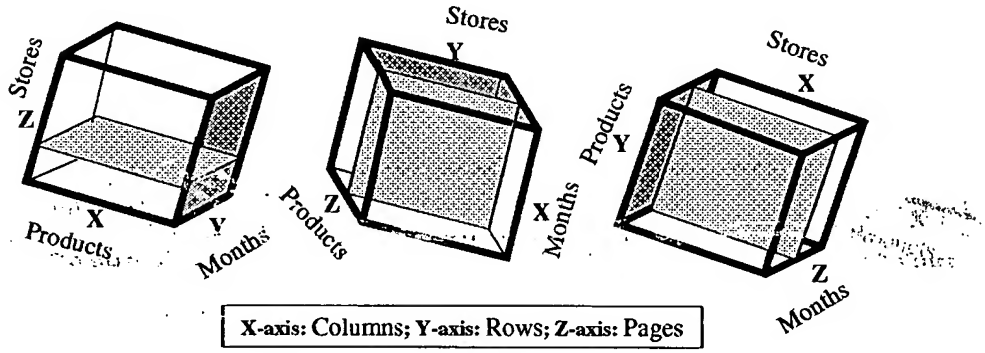
Figure 15-13 Three-dimensional display with roll-up.

page display on the spreadsheet. The columns represent the various products, the rows represent the months, and the pages represent the stores. At this point, if you want to roll up to the next higher level of subcategory, how will the display in Figure 15-6 change? The columns on the display will have to change to represent subcategories instead of products. Please see Figure 15-13 indicating this change.

Let us ask just one more question before we leave this subsection. When you have rolled up to the subcategory level in the product dimension, what happens to the display if you also roll up to the next higher level of the store dimension, territory? How will the display on the spreadsheet change? Now the spreadsheet will display the sales with columns representing subcategories, rows representing months, and the pages representing territories.

Slice-and-Dice or Rotation

Let us revisit Figure 15-6 showing the display of months as rows, products as columns, and stores as pages. Each page represents the sales for one store. The data model corresponds to a physical cube with these data elements represented by its primary edges. The page displayed is a slice or two-dimensional plane of the cube. In particular, this display page for the New York store is the slice parallel to the product and time axes. Now begin to look at Figure 15-14 carefully. On the left side, the first part of the diagram shows this alignment of the cube. For the sake simplicity, only three products, three months, and three stores are chosen for illustration.



Store: New York

	Hats	Coats	Jackets
Jan	200	550	350
Feb	210	480	390
Mar	190	480	380

Product: Hats

	Jan	Feb	Mar
New York	200	210	190
Boston	210	250	240
San Jose	130	90	70

Month: January

	New York	Boston	San Jose
Hats	200	210	130
Coats	550	500	200
Jackets	350	400	100

Figure 15-14 Slicing and dicing.

Now rotate the cube so that products are along the Z-axis, months are along the X-axis, and stores are along the Y-axis. The slice we are considering also rotates. What happens to the display page that represents the slice? Months are now shown as columns and stores as rows. The display page represents the sales of one product, namely product: hats.

You can go to the next rotation so that months are along the Z-axis, stores are along the X-axis, and products are along the Y-axis. The slice we are considering also rotates. What happens to the display page that represents the slice? Stores are now shown as columns and products as rows. The display page represents the sales of one month, namely month: January.

What is the great advantage of all of this for the users? Did you notice that with each rotation, the users can look at page displays representing different versions of the slices in the cube. The users can view the data from many angles, understand the numbers better, and arrive at meaningful conclusions.

Uses and Benefits

After exploring the features of OLAP in sufficient detail, you must have already deduced the enormous benefits of OLAP. We have discussed multidimensional analysis as provided in OLAP systems. The ability to perform multidimensional analysis with complex queries sometimes also entails complex calculations.

Let us summarize the benefits of OLAP systems:

- Increased productivity of business managers, executives, and analysts
- Inherent flexibility of OLAP systems means that users may be self-sufficient in running their own analysis without IT assistance
- Benefit for IT developers because using software specifically designed for the system development results in faster delivery of applications
- Self-sufficiency of users, resulting in reduction in backlog
- Faster delivery of applications following from the previous benefits
- More efficient operations through reducing time on query executions and in network traffic
- Ability to model real-world challenges with business metrics and dimensions

OLAP MODELS

Have you heard of the terms ROLAP or MOLAP? There is another variation, DOLAP. A very simple explanation of the variations relates to the way data is stored for OLAP. The processing is still online analytical processing, only the storage methodology is different.

ROLAP stands for relational online analytical processing and MOLAP stands for multidimensional online analytical processing. In either case, the information interface is still OLAP. DOLAP stands for desktop online analytical processing. DOLAP is meant to provide portability to users of online analytical processing. In the DOLAP methodology, multidimensional datasets are created and transferred to the desktop machine, requiring only the DOLAP software to exist on that machine. DOLAP is a variation of ROLAP.

Overview of Variations

In the MOLAP model, online analytical processing is best implemented by storing the data multidimensionally, that is, easily viewed in a multidimensional way. Here the data structure is fixed so that the logic to process multidimensional analysis can be based on well-defined methods of establishing data storage coordinates. Usually, multidimensional databases (MDDBs) are vendors' proprietary systems. On the other hand, the ROLAP model relies on the existing relational DBMS of the data warehouse. OLAP features are provided against the relational database.

See Figure 15-15 contrasting the two models. Notice the MOLAP model shown on the left side of the figure. The OLAP engine resides on a special server. Proprietary multidimensional databases (MDDBs) store data in the form of multidimensional hypercubes. You have to run special extraction and aggregation jobs to create these multidimensional data cubes in the MDDBs from the relational database of the data warehouse. The special server presents the data as OLAP cubes for processing by the users.

On the right side of the figure you see the ROLAP model. The OLAP engine resides on the desktop. Prefabricated multidimensional cubes are not created beforehand and stored in special databases. The relational data is presented as virtual multidimensional data cubes.

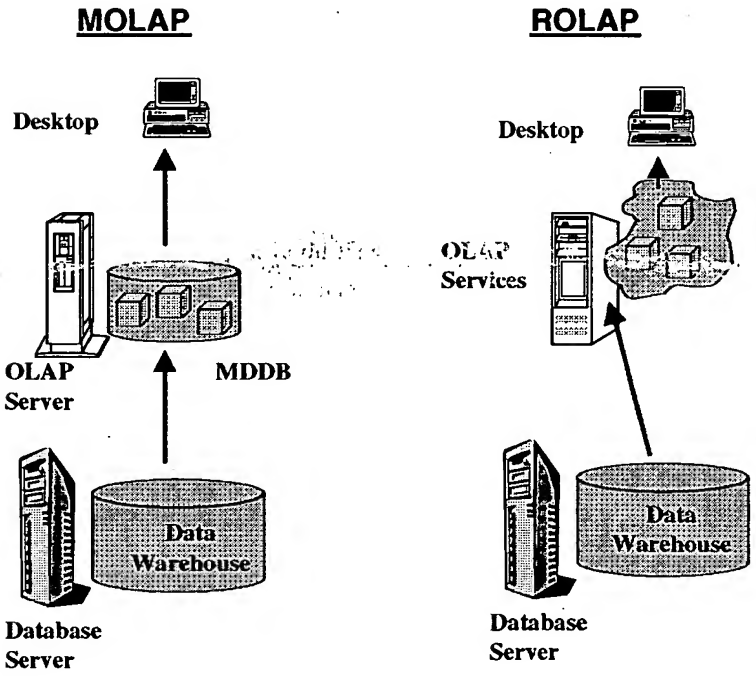


Figure 15-15 OLAP models.

The MOLAP Model

As discussed, in the MOLAP model, data for analysis is stored in specialized multidimensional databases. Large multidimensional arrays form the storage structures. For example, to store sales number of 500 units for product ProductA, in month number 2001/01, in store StoreS1, under distributing channel Channel05, the sales number of 500 is stored in an array represented by the values (ProductA, 2001/01, StoreS1, Channel05).

The array values indicate the location of the cells. These cells are intersections of the values of dimension attributes. If you note how the cells are formed, you will realize that not all cells have values of metrics. If a store is closed on Sundays, then the cells representing Sundays will all be nulls.

Let us now consider the architecture for the MOLAP model. Please go over each part of Figure 15-16 carefully. Note the three layers in the multitier architecture. Precalculated and prefabricated multidimensional data cubes are stored in multidimensional databases. The MOLAP engine in the application layer pushes a multidimensional view of the data from the MDDBs to the users.

As mentioned earlier, multidimensional database management systems are proprietary software systems. These systems provide the capability to consolidate and fabricate summarized cubes during the process that loads data into the MDDBs from the main data warehouse. The users who need summarized data enjoy fast response times from the preconsolidated data.

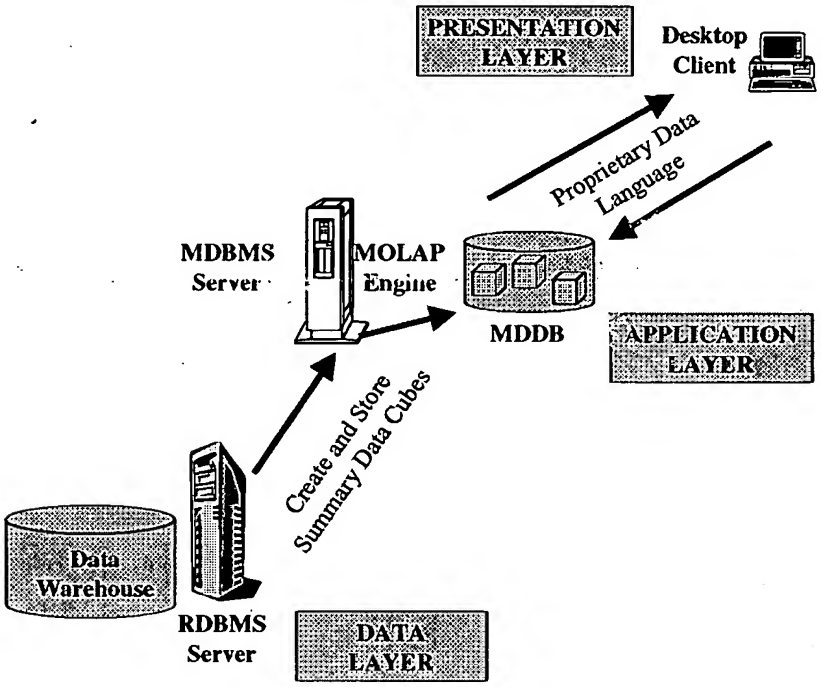


Figure 15-16 The MOLAP model.

The ROLAP Model

In the ROLAP model, data is stored as rows and columns in relational form. This model presents data to the users in the form of business dimensions. In order to hide the storage structure to the user and present data multidimensionally, a semantic layer of metadata is created. The metadata layer supports the mapping of dimensions to the relational tables. Additional metadata supports summarizations and aggregations. You may store the metadata in relational databases.

Now see Figure 15-17. This figure shows the architecture of the ROLAP model. What you see is a three-tier architecture. The analytical server in the middle tier application layer creates multidimensional views on the fly. The multidimensional system at the presentation layer provides a multidimensional view of the data to the users. When the users issue complex queries based on this multidimensional view, the queries are transformed into complex SQL directed to the relational database. Unlike the MOLAP model, static multidimensional structures are not created and stored.

True ROLAP has three distinct characteristics:

- Supports all the basic OLAP features and functions discussed earlier
- Stores data in a relational form
- Supports some form of aggregation

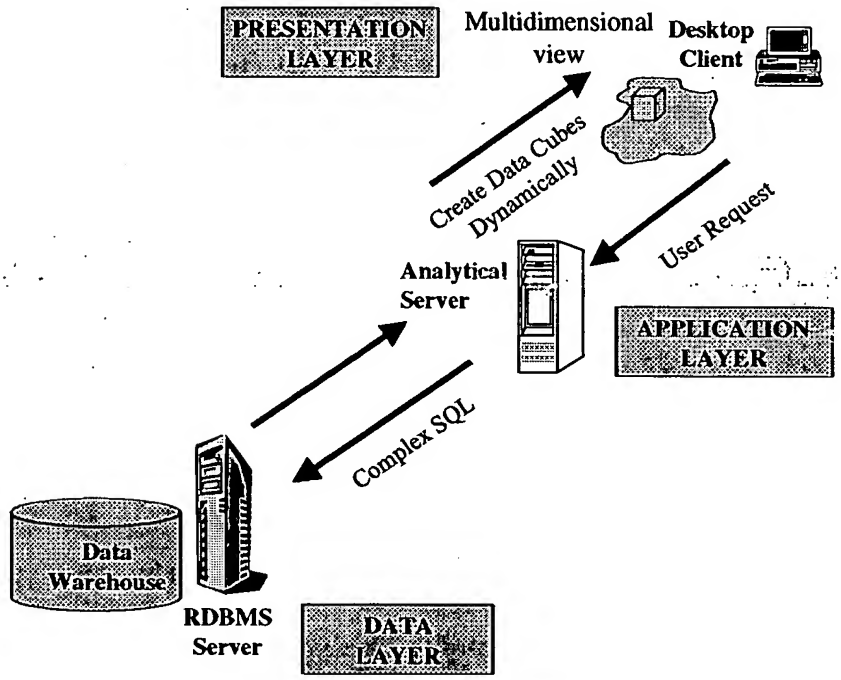


Figure 15-17 The ROLAP model.

Local hypercubing is a variation of ROLAP provided by vendors. This is how it works:

1. The user issues a query.
2. The results of the query get stored in a small, local, multidimensional database.
3. The user performs analysis against this local database.
4. If additional data is required to continue the analysis, the user issues another query and the analysis continues.

ROLAP VERSUS MOLAP

Should you use the relational approach or the multidimensional approach to provide online analytical processing for your users? That depends on how important query performance is for your users. Again, the choice between ROLAP and MOLAP also depends on the complexity of the queries from your users. Figure 15-18 charts the solution options based on the considerations of query performance and complexity of queries. MOLAP is the choice for faster response and more intensive queries. These are just two broad considerations.

As part of the technical component of the project team, your perspective on the choice is entirely different from that of the users. Users will get the functionality and benefits of multidimensionality from either model but are more concerned with questions relating to the extent of business data made available for analysis, the acceptability of performance, and the justification of the cost.

Let us conclude the discussion on the choice between ROLAP and MOLAP with Figure 15-19. This figure compares the two models based on the specific aspects of data storage, technologies, and features. This figure is important, for it pulls everything together and presents a balanced case.

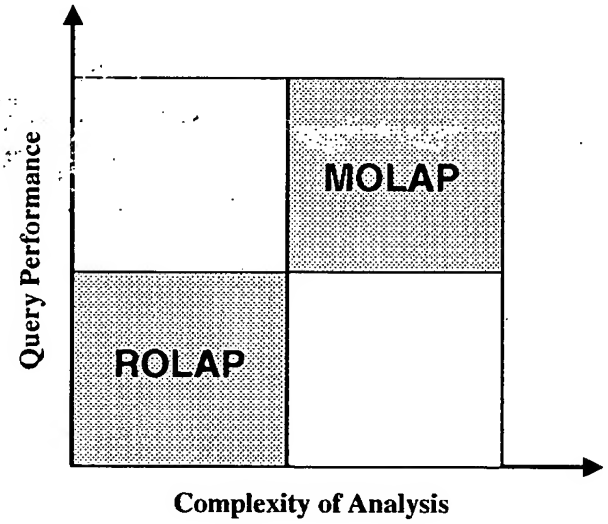


Figure 15-18 ROLAP or MOLAP?

	Data Storage	Underlying Technologies	Functions and Features
ROLAP	Data stored as relational tables in the warehouse. Detailed and light summary data available. Very large data volumes. All data access from the warehouse storage.	Use of complex SQL to fetch data from warehouse. ROLAP engine in analytical server creates data cubes on the fly. Multidimensional views by presentation layer.	Known environment and availability of many tools. Limitations on complex analysis functions. Drill-through to lowest level easier. Drill-across not always easy.
MOLAP	Data stored as relational tables in the warehouse. Various summary data kept in proprietary databases (MDDBs) Moderate data volumes. Summary data access from MDDB, detailed data access from warehouse.	Creation of pre-fabricated data cubes by MOLAP engine. Proprietary technology to store multidimensional views in arrays, not tables. High speed matrix data retrieval. Sparse matrix technology to manage data sparsity in summaries.	Faster access. Large library of functions for complex calculations. Easy analysis irrespective of the number of dimensions. Extensive drill-down and slice-and-dice capabilities.

Figure 15-19 ROLAP versus MOLAP.

OLAP IMPLEMENTATION CONSIDERATIONS

Before considering implementation of OLAP in your data warehouse, you have to take into account two key issues with regard to the MOLAP model running under MDDBMS. The first issue relates to the lack of standardization. Each vendor tool has its own client interface. Another issue is scalability. OLAP is generally good for handling summary data, but not good for volumes of detailed data.

On the other hand, highly normalized data in the data warehouse can give rise to processing overhead when you are performing complex analysis. You may reduce this by using a STAR schema multidimensional design. In fact, for some ROLAP tools, the multidimensional representation of data in a STAR schema arrangement is a prerequisite.

Consider a few choices of architecture. Look at Figure 15-20 showing four architectural options.

You have now studied the various implementation options for providing OLAP functionality in your data warehouse. These are important choices. Remember, without OLAP, your users have very limited means for analyzing data. Let us now examine some specific design considerations.

Data Design and Preparation

The data warehouse feeds data to the OLAP system. In the MOLAP model, separate proprietary multidimensional databases store the data fed from the data warehouse in the form of multidimensional cubes. On the other hand, in the ROLAP model, although no static intermediary data repository exists, data is still pushed into the OLAP system with

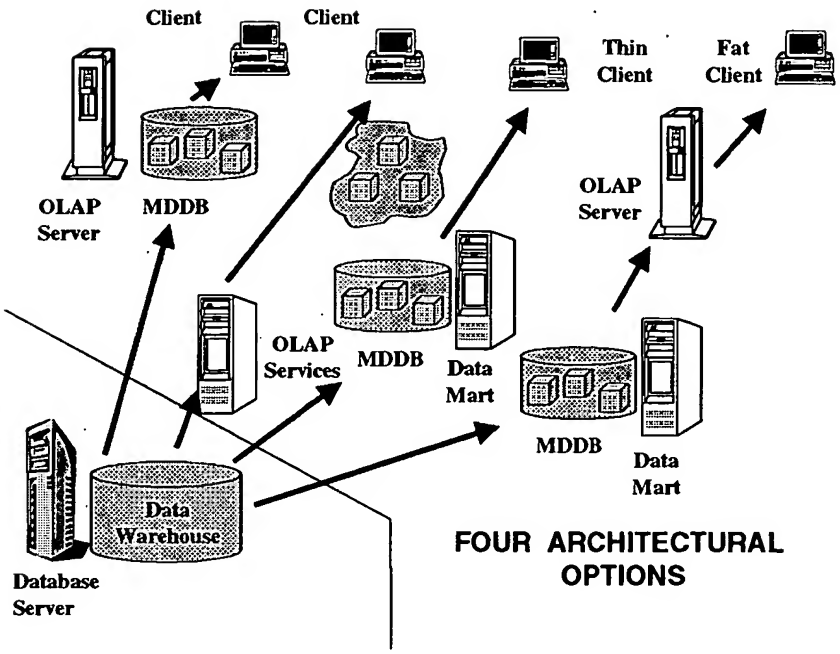


Figure 15-20 OLAP architectural options.

cubes created dynamically on the fly. Thus, the sequence of the flow of data is from the operational source systems to the data warehouse and from there to the OLAP system.

Sometimes, you may have the desire to short-circuit the flow of data. You may wonder why you should not build the OLAP system on top of the operational source systems themselves. Why not extract data into the OLAP system directly? Why bother moving data into the data warehouse and then into the OLAP system? Here are a few reasons why this approach is flawed:

- An OLAP system needs transformed and integrated data. The system assumes that the data has been consolidated and cleansed somewhere before it arrives. The disparity among operational systems does not support data integration directly.
- The operational systems keep historical data only to a limited extent. An OLAP system needs extensive historical data. Historical data from the operational systems must be combined with archived historical data before it reaches the OLAP system.
- An OLAP system requires data in multidimensional representations. This calls for summarization in many different ways. Trying to extract and summarize data from the various operational systems at the same time is untenable. Data must be consolidated before it can be summarized at various levels and in different combinations.
- Assume there are a few OLAP systems in your environment. That is, one supports the marketing department, another the inventory control department, yet another the finance department, and so on. To accomplish this, you have to build a separate interface with the operational systems for data extraction into each OLAP system. Can you imagine how difficult this would be?

In order to help prepare the data for the OLAP system, let us first examine some significant characteristics of data in this system. Please review the following list:

- An OLAP system stores and uses much less data compared to a data warehouse.
- Data in the OLAP system is summarized. You will rarely find data at the lowest level of detail as in the data warehouse.
- OLAP data is more flexible for processing and analysis partly because there is much less data to work with.
- Every instance of the OLAP system in your environment is customized for the purpose that instance serves. In order words, OLAP data tends to be more departmentalized, whereas data in the data warehouse serves corporate-wide needs.

An overriding principle is that OLAP data is generally customized. When you build the OLAP system with system instances servicing different user groups, you need to keep this in mind. For example, one instance or specific set of summarizations would be meant for one group of users, say the marketing department. Let us quickly go through the techniques for preparing OLAP data for a specific group of users or a particular department, for example, marketing.

Define Subset. Select the subset of detailed data the marketing department is interested in.

Summarize. Summarize and prepare aggregate data structures in the way the marketing department needs for summarizing. For example, summarize products along product categories as defined by marketing. Sometimes, marketing and accounting departments may categorize products in different ways.

Denormalize. Combine relational tables in exactly the same way the marketing department needs denormalized data. If marketing needs tables A and B joined, but finance needs tables B and C joined, go with the join for tables A and B for the marketing OLAP subset.

Calculate and Derive. If some calculations and derivations of the metrics are department-specific in your company, use the ones for marketing.

Index. Choose those attributes that are appropriate for marketing to build indexes.

What about data modeling for the OLAP data structure? The OLAP structure contains several levels of summarization and a few kinds of detailed data. How do you model these levels of summarization?

Please see Figure 15-21 indicating the types and levels of data in OLAP systems. These types and levels must be taken into consideration while performing data modeling for the OLAP systems. Pay attention to the different types of data in an OLAP system. When you model the data structures for your OLAP system, you need to provide for these types of data.

Administration and Performance

Let us now turn our attention to two important though not directly connected issues.

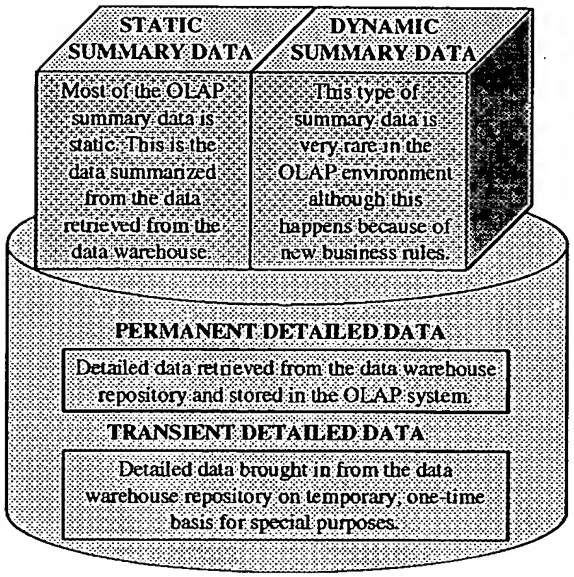


Figure 15-21 Data modeling considerations for OLAP.

Administration. One of these issues is the matter of administration and management of the OLAP environment. The OLAP system is part of the overall data warehouse environment and, therefore, administration of the OLAP system is part of the data warehouse administration. Nevertheless, we must recognize some key considerations for administering and managing the OLAP system. Let us briefly indicate a few of these considerations.

- Expectations on what data will be accessed and how
- Selection of the right business dimensions
- Selection of the right filters for loading the data from the data warehouse
- Methods and techniques for moving data into the OLAP system (MOLAP model)
- Choosing the aggregation, summarization, and precalculation
- Developing application programs using the proprietary software of the OLAP vendor
- Size of the multidimensional database
- Handling of the sparse-matrix feature of multidimensional structures
- Drill down to the lowest level of detail
- Drill through to the data warehouse or to the source systems
- Drill across among OLAP system instances
- Access and security privileges
- Backup and restore facilities

Performance. First you need to recognize that the presence of an OLAP system in your data warehouse environment shifts the workload. Some of the queries that usually must run against the data warehouse will now be redistributed to the OLAP system. The

types of queries that need OLAP are complex and filled with involved calculations. Long and complicated analysis sessions consist of such complex queries. Therefore, when such queries get directed to the OLAP system, the workload on the main data warehouse becomes substantially reduced.

A corollary of shifting the complex queries to the OLAP system is the improvement in the overall query performance. The OLAP system is designed for complex queries. When such queries run in the OLAP system, they run faster. As the size of the data warehouse grows, the size of the OLAP system still remains manageable and comparably small.

Multidimensional databases provide a reasonably predictable, fast, and consistent response to every complex query. This is mainly because OLAP systems preaggregate and precalculate many, if not, all possible hypercubes and store these. The queries run against the most appropriate hypercubes. For instance, assume that there are only three dimensions. The OLAP system will calculate and store summaries as follows:

- A three-dimensional low-level array to store base data
- A two-dimensional array of data for dimension-1 and dimension-2
- A 2-dimensional array of data for dimension-2 and dimension-3
- A high-level summary array by dimension-1
- A high-level summary array by dimension-2
- A high-level summary array by dimension-3

All of these precalculations and preaggregations result in faster response to queries at any level of summarization. But this speed and performance do not come without any cost. You pay the price to some extent in the load performance. OLAP systems are not refreshed daily for the simple reason that load times for precalculating and loading all the possible hypercubes are exhorbitant. Enterprises use longer intervals between refreshes of their OLAP systems. Most OLAP systems are refreshed once a month.

OLAP Platforms

Where does the OLAP system physically reside? Should it be on the same platform as the main data warehouse? Should it be planned to be on a separate platform from the beginning? What about growth of the data warehouse and the OLAP system? How do the growth patterns affect the decision? These are some of the questions you need to answer as you provide OLAP capability to your users.

Usually, the data warehouse and the OLAP system start out on the same platform. When both are small, it is cost-justifiable to keep both on the same platform. Within a year, it is usual to find rapid growth in the main data warehouse. The trend normally continues. As this growth happens, you may want to think of moving the OLAP system to another platform to ease the congestion. But how exactly would you know whether to separate the platforms and when to do so? Here are some guidelines:

- When the size and usage of the main data warehouse escalate and reach the point where the warehouse requires all the resources of the common platform, start acting on the separation.
- If too many departments need the OLAP system, then the OLAP requires additional platforms to run.

- Users expect the OLAP system to be stable and perform well. The data refreshes to the OLAP system are much less frequent. Although this is true for the OLAP system, daily application of incremental loads and full refreshes of certain tables are needed for the main data warehouse. If these daily transactions applicable to the data warehouse begin to disrupt the stability and performance of the OLAP system, then move the OLAP system to another platform.
- Obviously, in decentralized enterprises with OLAP users spread out geographically, one or more separate platforms for the OLAP system become necessary.
- If users of one instance of the OLAP system want to stay away from the users of another, then separation of platforms needs to be looked into.
- If the chosen OLAP tools need a configuration different from the platform of the main data warehouse, then the OLAP system requires a separate platform, configured correctly.

OLAP Tools and Products

The OLAP market is becoming sophisticated. Many OLAP products have appeared and most of the recent products are quite successful. Quality and flexibility of the products have improved remarkably.

Before we provide a checklist to be used for evaluation of OLAP products, let us list a few broad guidelines:

- Let your applications and the users drive the selection of the OLAP products. Do not be carried away by flashy technology.
- Remember, your OLAP system will grow both in size and in the number of active users. Determine the scalability of the products before you choose.
- Consider how easy it is to administer the OLAP product.
- Performance and flexibility are key ingredients in the success of your OLAP system.
- As technology advances, the differences in the merits between ROLAP and MOLAP appear to be somewhat blurred. Do not worry too much about these two methods. Concentrate on the matching of the vendor product with your users' analytical requirements. Flashy technology does not always deliver.

Now let us get to the selection criteria for choosing OLAP tools and products. While you evaluate the products, use the following checklist and rate each product against each item on the checklist:

- Multidimensional representation of data
- Aggregation, summarization, precalculation, and derivations
- Formulas and complex calculations in an extensive library
- Cross-dimensional calculations
- Time intelligence such as year-to-date, current and past fiscal periods, moving averages, and moving totals
- Pivoting, cross-tabs, drill-down, and roll-up along single or multiple dimensions

- Interface of OLAP with applications and software such as spreadsheets, proprietary client tools, third-party tools, and 4GL environments.

Implementation Steps

At this point, perhaps your project team has been given the mandate to build and implement an OLAP system. You know the features and functions. You know the significance. You are also aware of the important considerations. How do you go about implementing OLAP? Let us summarize the key steps. These are the steps or activities at a very high level. Each step consists of several tasks to accomplish the objectives of that step. You will have to come up with the tasks based on the requirements of your environment. Here are the major steps:

- Dimensional modeling
- Design and building of the MDDB
- Selection of the data to be moved into the OLAP system
- Data acquisition or extraction for the OLAP system
- Data loading into the OLAP server
- Computation of data aggregation and derived data
- Implementation of application on the desktop
- Provision of user training

CHAPTER SUMMARY

- OLAP is critical because its multidimensional analysis, fast access, and powerful calculations exceed that of other analysis methods.
- OLAP is defined on the basis of Codd's initial twelve guidelines.
- OLAP characteristics include multidimensional view of the data, interactive and complex analysis facility, ability to perform intricate calculations, and fast response time.
- Dimensional analysis is not confined to three dimensions that can be represented by a physical cube. Hypercubes provide a method for representing views with more dimensions.
- ROLAP and MOLAP are the two major OLAP models. The difference between them lies in the way the basic data is stored. Ascertain which model is more suitable for your environment.
- OLAP tools have matured. Some RDBMSs include support for OLAP.

REVIEW QUESTIONS

1. Briefly explain multidimensional analysis.
2. Name any four key capabilities of an OLAP system.
3. State any five of Dr. Codd's guidelines for an OLAP system, giving a brief description for each.

4. What are hypercubes? How do they apply in an OLAP system?
5. What is meant by slice-and-dice? Give an example.
6. What are the essential differences between the MOLAP and ROLAP models? Also list a few similarities.
7. What are multidimensional databases? How do these store data?
8. Describe any one of the four OLAP architectural options.
9. Discuss two reasons why feeding data into the OLAP system directly from the source operational systems is not recommended.
10. Name any four factors for consideration in OLAP administration.

EXERCISES

1. Indicate if true or false:
   A. OLAP facilitates interactive queries and complex uses.
   B. A hypercube can be represented by the physical cube.
   C. Slice-and-dice is the same as the rotation of the columns and rows in presentation of data.
   D. DOLAP stands for departmental OLAP.
   E. ROLAP systems store data in a multidimensional, proprietary databases.
   F. The essential difference between ROLAP and MOLAP is in the way data is stored.
   G. OLAP systems need transformed and integrated data.
   H. Data in an OLAP system is rarely summarized.
   I. Multidimensional domain structure (MDS) can represent only up to six dimensions.
   J. OLAP systems do not handle moving averages.
2. As a senior analyst on the project team of a publishing company exploring the options for a data warehouse, make a case for OLAP. Describe the merits of OLAP and how it will be essential in your environment.
3. Pick any six of Dr. Codd's initial guidelines for OLAP. Give your reasons why the selected six are important for OLAP.
4. You are asked to form a small team to evaluate the MOLAP and ROLAP models and make your recommendations. This is part of the data warehouse project for a large manufacturer of heavy chemicals. Describe the criteria your team will use to make the evaluation and selection.
5. Your company is the largest producer of chicken products, selling to supermarkets, fast-food chains, and restaurants, and also exporting to many countries. The analysts from many offices worldwide expect to use the OLAP system when implemented. Discuss how the project team must select the platform for implementing OLAP for the company. Explain your assumptions.

Software Quality Journal 7, 179–193 (1998)

Typology of database quality factors

JOHN A. HOXMEIER

Computer Information Systems Department, College of Business, Colorado State University, Fort Collins, CO 80523, USA, jhox@lamar.colostate.edu

Databases are a critical element of virtually all conventional and ebusiness applications. How does an organization know if the information derived from the database is any good? To ensure a quality database application, should the emphasis during model development be on the application of quality assurance metrics (designing it right)? A large number of database applications fail or are unusable. A quality process does not necessarily lead to a usable database product. A database application can also be 'well-formed' with high data quality but lack semantic or cognitive fidelity (the right design). This paper expands on the growing body of literature in the area of data quality by proposing additions to a hierarchy of database quality dimensions that includes model and behavioral factors in addition to process and data factors.

Keywords: software quality; database quality; data quality; data models

Introduction

The ultimate objective of database analysis, design, and implementation is to establish an electronic repository that is a physical and behavioral model of the manageable aspects of a user's information domain. Database design is a complex, complicated art. Many complex factors must be considered during the process including, but not limited to, historical and future information requirements, the diversity of the data consumer community, organizational requirements, security, cost, ownership, performance, interface issues, and data integrity. These factors contribute to the success of a database application in both quantitative and qualitative ways and determine the overall quality of the database application. *Process* and *data* quality is quantitative management factors that are fairly well documented and understood, albeit underutilized. However, data *model* and *behavioral* considerations include important qualitative factors that contribute to overall database quality. A database is more than the instances of the data it manages. Data quality, while important, is just one element of assessing overall database quality.

This paper expands on the growing body of literature in the area of data quality by proposing additions to a hierarchy of database quality dimensions that includes model and behavioral factors in addition to the process and data factors. The term "database quality" in this context expands on the ISO definition of quality, i.e. *conformance to requirements* and *fitness for use* (1993). This definition is not adequate for the purposes of assessing database quality. While the requirement definition phase of the system development life cycle is critical to the success of an application, doing a good job of defining requirements is not sufficient in the implementation of a successful database

application. A database must also be judged by how closely it represents the world of the data consumer (the model), its ability to respond to both routine and unanticipated requests within the domain it is expected to manage (the behavior), and maintain this representation over time. The framework presented herein expands on work previously proposed (Hoxmeier, 1997; Hoxmeier and Monarchi, 1996) and incorporates data quality dimensions put forth by several prominent data quality researchers (Ballou and Pazar, 1995; Storey and Wang, 1994; Strong et al., 1997; Orr, 1998; Wand and Wang, 1996; Wang, 1998; Wang et al., 1993; Wang et al., 1995; Wang et al., 1999).

The problem/solution cycle

The database design process is largely driven by the requirements and needs of the consumer, who establishes the boundaries and properties of the problem domain and the requirements of the information. As organizations seek to preserve organizational memory and manage richer forms of information over broader networks, this task has become increasingly more difficult.

Figure 1 illustrates a typical scenario in the problem to solution cycle. It is not difficult to see why so many database applications are ultimately unsuitable to the consumer. Designers attempt to conceptualize the problem domain into a suitable physical implementation. The proposed solution is subject to many constraints including

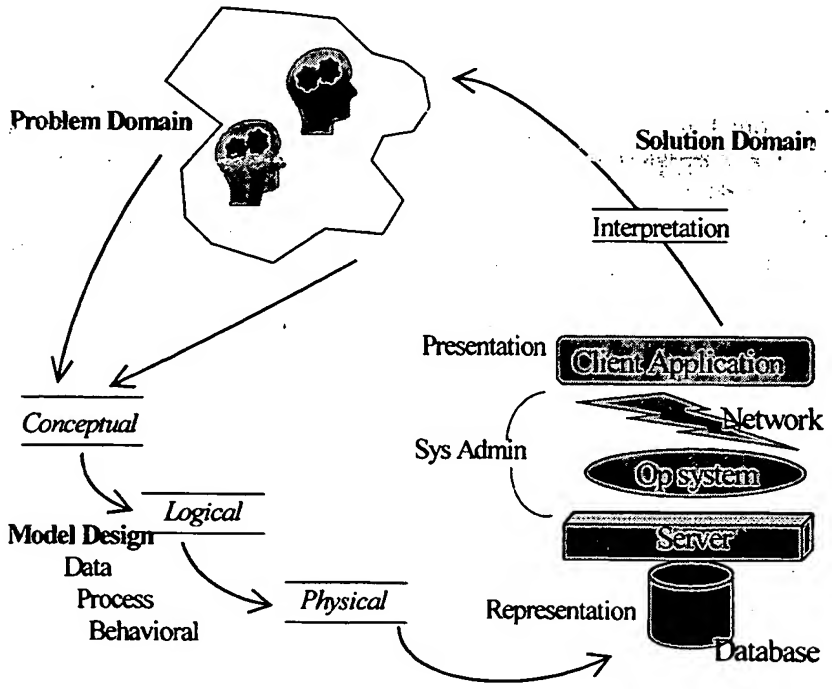


Fig. 1. Problem to solution cycle

the physical representation, system administration, application presentation, and information interpretation. These constraints or solution layers all contribute to the perceived quality of the solution by the information consumer. Figure 1 also shows the critical elements in the problem to solution cycle that are the bases for the discussion on database quality dimensions:

- The cycle *process* must be managed toward a successful outcome.
- The *model* itself must represent a usually diverse and fuzzy problem domain.
- The quality of the *data* in the database must be of sufficient grade.
- The application must *behave* in a way the consumer understands.

The last step depicted in the illustration, interpretation, is probably outside of the direct control of the design and development team. However, the consumer's ability to interpret the information is also critical to the success of a database application and, therefore, to the perceived quality of the database.

To ensure a quality database application, should the emphasis during model development be on the application of quality assurance metrics (designing it right)? It's hard to argue against this point, but there are a significant number of studies and anecdotal evidence that suggests that a large number of database applications fail, are unusable, or contribute to negative organizational consequences (Standish Group, 1997; Redman, 1998; Wand and Wang, 1996). A quality process does not necessarily lead to a usable database product (Arthur, 1997; Hoxmeier, 1995; Redman, 1995). A database should be evaluated in production based on certain quantitative and information-preserving transformation measures, such as data quality, data integrity, normalization, and performance. However, there are also many examples of database applications that are in most ways 'well-formed' with high data quality but lack semantic or cognitive fidelity (the right design). Additionally, determining and implementing the proper set of database behaviors can be an elusive task. Depending on the risk factors affecting the application, there may be certain aspects of the quality assessment that deserve heavier weights. Contrary to the popular notion of product quality, whether the database meets the expectations of its end-users is only one aspect of overall database quality.

Significant prior research

Quality metrics have been used for years in the design, development, and marketing for consumer goods and services. Quality engineering methods, such as Total Quality Management (TQM) and Quality Function Deployment (QFD) are commonly used by many product design and manufacturing disciplines, and are rapidly entering the service disciplines. In the area of information quality, however, the use of these techniques is virtually non-existent. Recently, researchers have begun to evaluate and study the characteristics of information as they would any other product or service (Kahn and Strong, 1998; Kaplan et al., 1998; Wang et al., 1995).

Researchers and practitioners alike have tried to establish a set of factors, attributes, rules or guidelines in order to evaluate system quality. Zmud concluded that a set of four dimensions divided into 25 factors represented the dimensions of information quality

(Zmud, 1978). The dimensions included data quality, relevancy, format quality, and meaning quality. Referring to information systems, James Martin stated that the collection of data has little value unless the data are used to understand the world and prescribe action to improve it (Martin, 1976). Martin proposed 12 qualities that computer-provided information should possess.

Cap Gemini Pandata, a Dutch company, uses a framework that decomposes the entire information quality notion into four dimensions, 21 aspects, and 40 attributes (Delen and Rijsenbrij, 1992). Cap Gemini has adopted this framework on the company procedures covering software package auditing. AT & T is researching data quality and have identified four primary factors including accuracy, currentness, completeness and consistency (Fox et al., 1994). Another group, the Southern California Online Users Group (SCOUG), defined characteristics of a quality library online database (Tenopir, 1990). The purpose of the set of characteristics was to allow professional searchers to rate each library online database system.

Marketing research has identified approaches used to assess product quality attributes that are important to consumers (Churchill, 1991; Menon, 1997). Wang et al. applied this concept toward a data consumer (1996). They performed a comprehensive survey that identified 4 high-level categories of data quality after evaluating 118 variables. The Wang et al. factors include intrinsic data quality, contextual data quality, representation data quality, and accessibility data quality. A recent study applied the model to a series of field studies that focused on the concerns of the data consumer (Strong et al., 1997). These field studies confirmed the dimensions of data quality set forth in the Wang study.

There appear to be many similarities in the factors identified in these studies based on the perspective of the evaluators. Both developers and data consumers are concerned with data quality metrics like accuracy, timeliness, consistency, etc. Most of the research, while focused on data or information quality, indicates that there is a diverse set of factors influencing data quality. Any individual variable however, such as accuracy, is difficult to quantify. Nonetheless, researchers have developed a fairly consistent view of data quality. There is little available in the literature on the evaluation of overall database quality including other considerations such as semantic fidelity, behavioral, and value factors.

The proposed framework

It is proposed that through the hierarchical framework presented in Fig. 2, one can evaluate overall database quality by assessing four primary dimensions: process, data, model, and behavior. Portions of the hierarchy draw heavily from previous studies on data and information quality, and documented process quality standards (Arthur, 1997; Wang, 1998). A dimension is a set of database quality attributes or components that most data consumers react to in a fairly consistent way (Wang et al., 1996). The use of a set of dimensions to represent a quality typology is consistent with previous quality research (Dvir and Evans, 1996; Wang et al., 1996; Strong et al., 1997). The framework presents the four dimensions in a dimension-attribute-property hierarchy.

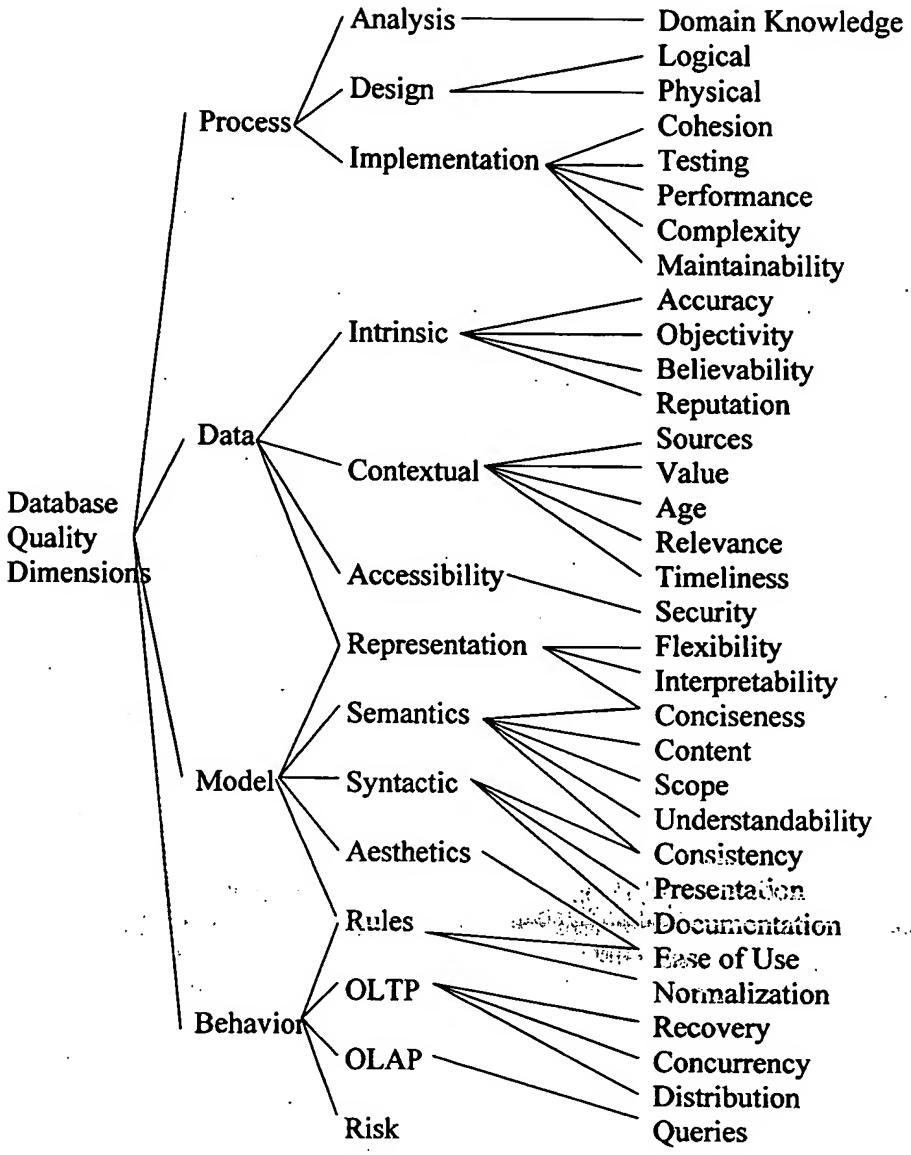


Fig. 2. Database quality dimensions

Process quality

Much attention has been given over the years to process quality improvement. *ISO-9000-3, Total Quality Management (TQM), Quality Function Deployment (QFD), and Capability Maturity Model (CMM)* are approaches that are concerned primarily with the incorporation of quality management within the process of systems development (Costin,

1994; Dvir and Evans, 1996; Herbsleb, 1997; Schmauch, 1994). *Quality control* is a process of ensuring that the database conforms to predefined standards and guidelines using statistical quality measures (Dyer, 1992). It compares variations of identified activities with the results of predetermined standards and assesses the variation between the two. When deviations from the problem domain are found, they are resolved and the process is modified as needed. This is an effective, yet reactive form of quality management. *Quality assurance* attempts to maintain the quality standards in a proactive way. In addition to using quality control measures, quality assurance goals go further by surveying the customer to determine their level of satisfaction with the product. Conceivably, potential problems can be detected early in the process.

The philosophy of ISO-9000-3 and CMM are to build quality into a software system on a continuous basis, from conception through implementation. ISO-9000-3 as a process quality standard does not offer any particular metrics to be utilized during the process. In addition, as a general software standard, ISO-9000-3 does not deal specifically with database issues. CMM is best regarded as a tool to be used to pursue an organization's business goals. The time and cost of a CMM-based software process improvement often exceeds the expectations of those involved.

A specific property addition to the framework within the dimension of process implementation quality is performance. All too often, specific performance requirements are either ignored during the design process or evaluated after implementation. While performance may be viewed by some as an implementation issue, it should be considered an important factor in overall database quality, even in the conceptual phase. Both relational and object databases can contain rather serious problems in terms of data redundancy, relationships, integrity, and structure. The objective is to design a normalized, high-fidelity database while minimizing complexity. When evaluating performance there are times when de-normalization may represent an optimal solution. However, anytime a general-purpose database is optimized for a given situation, other requirements inevitably arise that negate the advantage. The measures used to assess the trade-off may include query and update performance, storage, and the avoidance of data anomalies. Similar to the contrast between data and semantic quality, a database that is otherwise well designed but does not perform well is useless.

Database data quality

Data integrity is one of the keys to developing a quality database. Without accurate data, users will lose confidence in the database or make uninformed decisions (Redman, 1995). While data integrity can become a problem over time, there are relatively straightforward ways to enforce constraints and domains and to ascertain when problems exist (Moriarty, 1996). The identification, interpretation, and application of business rules, however, present a more difficult challenge for the developer. Rules and policies must be communicated and translated and much of the meaning and intent can be lost in this process. Because data quality has been a focus of previous research (for an excellent discussion, see Strong et al., 1997 or Wang et al., 1999) and these studies have been used as a basis for the data dimension presented here, the individual attributes will not be discussed. However, a couple of additional properties are worth noting.

A frequently overlooked metric in the evaluation of data integrity is the age of the data, database, and model. Data or model age is different than the timeliness property. Timeliness refers to the delay between availability and accessibility. Age refers to the time that has passed since the data was entered into the database or when the data model was developed. The data should only be as old as the problem domain and information sources will allow and maintained only as long as the situation requires. This can be a few seconds or several years. At some point, the data needs to be refreshed in order to maintain its currency. Over time, the age of the model may degrade in its ability to depict the problem domain. The model must be updated so that as the problem domain changes, the model of the database changes as well.

Additionally, the assessment of data quality must include value considerations. Time and financial constraints are real concerns. As IT departments are expected to do more with less and as cycle times continue to decrease for database applications, developers must make decisions about the extent to which they are going to implement and evaluate quality considerations. Shorter cycle times present a good argument for modularity and reusability, so quality factors must be addressed on a micro basis.

Data model quality

As has been presented, data quality is usually associated with the quality of the data values. However, even data that meets all other quality criteria is of little use if it is based on a deficient data model (Levitin and Redman, 1995). Data model quality is the third of the four high level dimensions presented above. Information and an application that represent a high proportionate match between the problem and solution domains should be the goal of a database with high semantic quality. Representation, semantics, syntax, and aesthetics are all attributes of model quality (Hoxmeier and Monarchi, 1996; Levitin and Redman, 1995; Lindland et al., 1994).

The database design process is largely driven by the requirements and needs of the data consumer, who establishes the boundaries and properties of the problem domain and the requirements of the task. The first step in the process, information discovery, is one of the most difficult, important, and labor intensive stages of database development (Chignell and Parsaye, 1993; Sankar and Marshall, 1993). It is in this stage where the semantic requirements are identified, prioritized, and visualized. Requirements can rarely be defined in a serial fashion. Generally, there is significant uncertainty over what these requirements are, and they only become clearer after considerable analysis, discussions with users, and experimentation with prototypes. This means previous work may be revisited. Additionally, while many studies point to the importance of user involvement in the discovery and design phase, many information consumers are uncertain about their requirements or have insufficient database knowledge to provide much insight.

Concentric design is an approach that is appropriate in database design. This cyclical process emulates the philosophy of continuous quality improvement used in TQM and CMM (Braithwaite, 1994; Dvir and Evans, 1994; Herbsleb, 1997). The costs associated with developing quality into the application from design to implementation are much lower than the costs of correcting problems that occur later due to poor design. However, the learning curve within the domain for the designer may be steep and the

demand for the application may force rapid delivery. So, how do designers arrive at high semantic quality in a very short period of time?

Qualitative techniques address the ambiguous and subjective dimensions of conceptual database design. The interaction between people and information is one where human preference and constraints have a huge impact on the effectiveness of database design. The use of techniques such as affinity and pareto diagrams, semantic object models, group decision support systems, nominal group, and interrelationship diagraphs help to improve the process of problem and solution domain definition. Well studied quantitative techniques, such as entity-relationship diagrams, object models, data flow diagrams, and performance benchmarks, on the other hand, allow the results of the qualitative techniques to be described in a visual format and measured in a meaningful way. Other object attributes that explicitly express quality can be included in the model as well. Storey and Wang present an innovative extension to the traditional ER approach for incorporating quality requirements (database quality data and product quality data) into conceptual database design (1994). The underlying premise of the approach is that quality requirements should be distinct from other database properties.

Qualitative and quantitative techniques can be used to assist the developer to extract a strong semantic model. However, it is difficult to design a database with high semantic value without significant domain knowledge and experience (Navathe, 1997). These may be the two most important considerations in databases of high semantic quality. In addition, conceptual database design remains more of an art than a science. It takes a high amount of creativity and vision to design a solution that is robust, usable, and can stand the test of time.

Database behavior quality

Many databases are perceived to be of low quality simply because they are difficult to use. In a survey in the UK, managers and professionals from various disciplines were asked to evaluate the quality of information they were using (Rolph and Bartram, 1994). Using 8 factors, "accuracy" rated the highest, "usable format" the lowest. Developers tend to focus on aspects of data quality at the expense of behavioral quality. Granted, the behaviors associated with a general-purpose database used for decision and analytical support are varied and complex.

What constitutes a database of high behavioral quality? Are the criteria different than those used for software applications in general? Clearly the behaviors for a database that is used to support transaction processing (OLTP) are different than those of a database used to support analytical processing (OLAP). Software development, in general, is very procedure- or function-driven. The objective is to build a system that works (and do it quickly). Database development, on the other hand, should be more focused on the content, context, behavior, semantics, and persistence of the data. Rapid application development and prototyping techniques contribute to arriving at a close match between the problem and solution domains. There may be no substitute for experience and proficiency with the software and tools used in the entire development process. It is one thing to discuss how a database should behave and even document these behaviors completely. Implementation and modification of these behaviors is an altogether different issue. The process of behavior implementation consists of the design and construction of a solution following the identification of the problem domain and the data model.

Because of the difficulties associated with the definition of a fixed set of current requirements and the determination of future utilization, the database problem domain is typically a moving target represented by the polygon in Fig. 3. The size and shape are constantly changing. In addition, insufficient identification of appropriate database 'behaviors', poor communication, and inexperience in the problem domain leads to inferior solutions. As a result, the solution domain rarely approaches the optimal solution presented in Fig. 3C. The database developer must attempt to develop a database model that closely matches the perceptions of the consumer, and deliver a design that can be implemented, maintained, and modified in a cost-effective way. A partial solution, such as that indicated in 3B, is more likely. The consumer will then dictate whether there is 1) enough of a solution to use, 2) the solution is of sufficient quality and, 3) whether they trust the database. Additionally, databases to be used in online analytical processing, data warehousing, or data mining applications present difficult challenges. The information consumer in these areas generally does not know what may be asked of the database. The database must behave in a fashion to respond to the most difficult requirement of all, that which the consumer has not yet thought of.

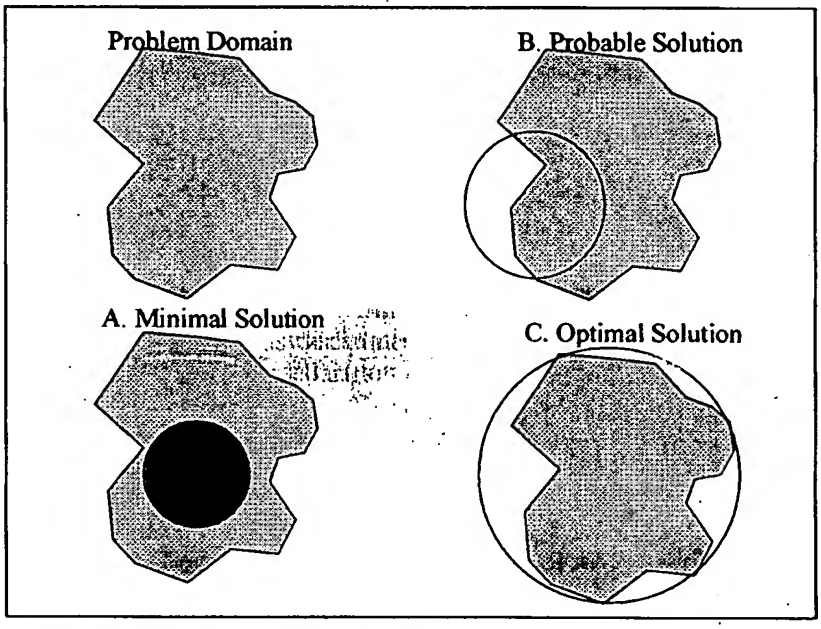


Fig. 3. Solution domain to problem mapping

And finally, an additional important contributor to database quality that is difficult to categorize is that of information risk. Risk is addressed in the project management literature but not discussed in the information quality literature. Risk may determine the grade of acceptable information quality. Consumers of on-line critical care database information that monitors hospital patients require a very high grade of information quality because the risk is very high. A database that tracks responses to a customer satisfaction survey, on the other hand, may be of lower grade because the overall information risk is low.

Case study

The framework was used as a basis to assess the overall quality of a production database for a large manufacturer of printers and scanners. The database is classified as an on-line analytical processing application in that it is used for decision support and is not part of the organization's mission critical OLTP applications. The database maintains information accumulated from the typical product warranty registration process. It includes product, demographic, purchase process, and customer profile information. The data was captured via an OCR scan and electronically loaded into the database. The data comes from all over the world and at the time of the review, represented well over 200,000 customers and over 2,000,000 warranty card responses. The marketing department originally articulated several objectives for the database:

- Capture important sales statistics for the product line as a basis for positioning future products.
- Track the purchase location for each of the products to quantify channel decisions.
- Establish a profile of customer information for future customer segmentation and product positioning.
- Report units sold by geographic region, product line, segment, period, and card question.

The system users within the marketing department became dissatisfied with the database based on several shortcomings that were contributing to a perceived lack of quality. Their concerns included:

- The information reported in the cross-tab type reports was not accurate and difficult to verify.
- The process for updating the database with new registrations was cumbersome and time-consuming.
- The department was unable to select a subset of the data for further analysis, e.g. list all customers who purchased their scanner through a particular retail store in the United States.
- The reports that were generated by the database took over 24-hours to produce. If the process was unsuccessful, the entire report had to be run again.

These were just a few of the concerns that led the company to initiate a review and potential redesign of the database. The company had invested a significant amount of time and resources on the system and was on the verge of dropping the project. The framework presented herein was used as a methodology for evaluating the database application and making suggestions for possible improvement. The four primary dimensions of process, data, model, and behavior quality were evaluated. Each will be briefly discussed below:

Process quality

The process that was deployed to develop the database was probably inadequate. The company had used a developer from outside the company because their internal IT group could not respond to their request for service. The consultant had developed the system based on very little interaction with the company. The contractor had significant

experience in the hardware and software architecture used by the company and database management system targeted for the application. However, the contractor had never designed and implemented a marketing application and the lead designer had limited formal relational training. While the implementation had gone smoothly, the contractor had not tested the database under loaded conditions and left the company with limited technical or design specifications. After interviewing company personnel about the process, they indicated that the contractor and never gone through a conceptual data model with them.

Data quality

There were several problems with the quality of the data in the database. Data quality was assessed using the information quality assessment (IQA) designed Wang & Strong (1999). This instrument uses 65 data quality assessment items to measure 15 different data quality variables. The scale for each item ranges from zero to ten where zero is labeled "not at all" and ten is labeled "completely." Questions measured the data quality variables of believability, accessibility, completeness, etc. This analysis indicated many data quality problems:

- The domain concept had not been implemented, so there were many inconsistencies within the attributes. For example, a query against the state attribute for the customer's address led to 61 distinct values. This type of variance was found in virtually every attribute. As a result, the reports produced combinations of crosstabs that were impossible to interpret.
- Because of the OCR process, many records had unrecognized characters and, as a result, were rejected during the acquisition process and not even included in the importation.
- Because of an inadequate design, many columns were duplicated, leading to erroneous totals.
- The time between initial data acquisition and final report aged the data to a point where it was unreliable.
- The consumers of the information had lost confidence in the data and overall believability was low.

Model quality

The contractor had not produced a conceptual data model of the application so it was difficult to assess the semantic match between the model produced by the contractor and the problem domain. However, an assessment of the logical and physical models revealed an impedence mismatch. The basic relationship that existed between the customer and the product they purchased was not represented in the database. The database should have shown a one to many to many between customers, products, and warranty responses. Rather, the physical implementation of the database repeated information on the customer for each product that the customer purchased. The individual survey questions were stored as attributes in a very long row. There were many null values and it was difficult to analyze the questions that were marked, "Please

check all that apply." This led to many data integrity and duplication problems that SQL simply cannot account for.

Application behavior quality

The application presented a user interface that was graphically appealing and relatively easy to use but lacked certain behaviors that were necessary to capture a smooth workflow. For example, it was possible to run the analytical reports for a particular period of time without having the data loaded in the database for that time period. In addition, because the primary key was automatically sequenced, it was possible to load the same data twice without knowing it. The physical indices that should have prevented such an occurrence were not present. Other application behaviors that were not present or insufficient included:

- The data was electronically loaded so there was no manual data entry. The forms for manually editing the data did not utilize any technique to enforce domain controls.
- The dates and times of reports and data changes were not recorded making audits difficult.
- The database had insufficient concurrency controls.
- The reports took several hours to produce and there was no status indicator.

Prescriptions

This database application suffered from several real and perceived deficiencies that contributed to its poor quality. The problems went beyond those associated with data quality alone. The process could have been improved and the project could have been better managed. Process improvement alone may not have been sufficient to improve the quality of the database itself. After all four areas of database quality were evaluated and considering the original objectives of the database, this customer registration application ranked very low for overall quality. Simply addressing the data quality problems would not have improved its utility. Several suggestions were made, all with varying associated costs. The following modifications were recommended with the applicable framework factor shown in parentheses:

- Redesign the basic data model to accurately reflect the problem domain. (model)
- Conform to data normalization strategies for relational tables. (model)
- Perform a series of data cleansing queries on each attribute after identifying the domain set. (data)
- Construct and implement domain enforcement within the database. (data)
- Add additional attributes for date and time stamping the rows. (data)
- Modify the user interface to encourage the user to follow a workflow. (behavior)
- Modify the application to protect against concurrency and update issues. (behavior)
- Add features to the forms to make it easy for the user to modify data using pre-approved domains. (behavior and data)
- Add physical indices to the database to prevent redundant information and improve performance. (process)

- Redesign the reports to represent the problem domain specified by the consumers. (behavior)
- Deploy the reports on the company intranet to reduce time and paper. (process)
- Summarize the results and highlight extraordinary or exceptional areas for the consumer. (data)
- Add a new layer to the user interface that makes it possible for the consumer to select certain areas for further analysis. (behavior)

Conclusion and research directions

How does one ensure a final database product that is of high quality? Database quality must be measured in terms of a combination of dimensions including process and behavior quality, data quality, and model fidelity. The framework presented above offers a typology for assessing these dimensions. The purpose of this paper was to expand on the existing research on data and process quality in an attempt to provide a more comprehensive view of database quality. The area is of great concern as information is viewed as a critical organizational asset and preserving organizational memory becomes a high priority (Saviano, 1997). A test case was shown as an example of an application of the framework, but further research is required to continue to validate the framework; to identify additional quality dimensions and develop metrics to quantify the properties; and to develop and deploy techniques to improve the fidelity of the data model.

References

L. Arthur. Quantum improvements in software system quality. *Communications of the ACM*, 40(6), (1997), 47–52.

D. Ballou and H. Pazer. Designing information systems to optimize the accuracy timeliness tradeoff. *Information Systems Research*, 6(1), (1995), 51–72.

T. Braithwaite. *Information Service Excellence Through TQM, Building Partnerships for Business Process Reengineering and Continuous Improvement*. (ASQC Quality Press, 1994).

M. Chignell and K. Parsaye. *Intelligent Database Tools and Applications*. (Wiley, California, 1993).

G.A. Churchill. *Marketing Research: Methodological Foundations*. (Dryden Press, 1991).

H. Costin. *Total Quality Management*. (Dryden, United States, 1994).

G. Delen and D. Rijsenbrij. A specification, engineering and measurement of information systems quality. *Journal of Systems Software*, 17(3), (1992), 205–217.

R. Dvir and S. Evans. A TQM Approach to the Improvement of Information Quality. Online , http: wem.mit.edu tdqm papers.html 1997, June , 1996.

M. Dyer. *The Cleanroom Approach to Quality Software Development*. (Wiley, 1992).

C. Fox, A. Levitin and T. Redman. The notion of data and its quality dimensions. *Information Processing and Management*, 30(1), (1994), 9–19.

J. Herbsleb, D. Zubrow, D. Goldenson, W. Hayes and M. Paulk. Software quality and capability maturity model, *Communications of the ACM*, 40(6), (1997), 30–40.

J. Hoxmeier. A framework for assessing database quality. *Proceedings of the Workshop on Behavioral Models and Design Transformations: Issues and Opportunities in Conceptual Modeling, ACM Sixteenth International Conference on Conceptual Modeling*, Los Angeles, CA, November, 1997.

J. Hoxmeier. Managing the legacy systems reengineering process: lessons learned and prescriptive advice. *Proceedings of the Seventh Annual Software Technology Conference*, Ogden ALC TISE, Salt Lake City, April, 1995.

J. Hoxmeier and D. Monachi. An assessment of database quality: design it right or the right design? *Proceedings of the Association for Information Systems Annual Meeting*, Phoenix, AZ, August, 1996.

ISO, International Organization for Standardization. *Quality-Vocabulary (Draft International Standard 8402)*. ISO Press, Geneva, Switzerland, 1993.

B. Kahn and D. Strong. Product and service performance model for information quality, *Proceedings of the 1998 Conference on Information Quality*, Cambridge, MA, 1998, 102–115.

D. Kaplan, R. Krishnan, R. Padman and J. Peters. Assessing data quality in accounting information systems. *Communications of the ACM*, 41(2), (1998), 72–78.

A. Levitin and T. Redman. Quality dimensions of a conceptual view. *Information Processing and Management*, 31(1), (1995).

O. Lindland, G. Sindre and A. Solvberg. Understanding quality in conceptual modeling. *IEEE Software*, Vol. 11, No. 2, March 1994, pp. 42–49.

J. Martin. *Principles of Database Management.* (Prentice-Hall, Inc., New Jersey, 1976).

A. Menon, B. Jaworski and A. Kohli. Product quality: impact of interdepartmental interactions. *Journal of the Academy of Marketing Science*, 25(3), (1997).

T. Moriarty. Barriers to data quality. *Database Programming and Design*, 61, May, 1996.

S. Navathe. Conceptual modeling in biomedical science. *Proceedings of the ACM Entity Relationship 97 Modeling Preconference Symposium*, Los Angeles, CA, 1997.

K. Orr. Data quality and systems theory. *Communications of the ACM*, 41(2), (1998), 66–71.

T.C. Redman. Improve data quality for competitive advantage. *Sloan Management Review*, 36(2), (1995), 99–107.

T.C. Redman. The impact of poor data quality on the typical enterprise. *Communications of the ACM*, 41(2), (1998), 79–82.

P. Rolph and P. Bartram. *The Information Agenda: Harnessing Relevant Information in a Changing Business Environment.* (Management Books 2000, London, 1994), pp. 65–87.

C. Sankar and T. Marshall. Database design support: an empirical investigation of perceptions and performance. *Journal of Database Management*, 4(3), (1993), 4–14.

J. Saviano. Are we there yet?. *CIO*, 87–96, June 1, 1997.

C. Schmauch. *ISO-9000 for Software Developers.* (ASQC Quality Press, 1994).

Standish Group. The Chaos Report. Online , http: www.standishgroup.com chaos.html 1997, November 7 , 1997.

V. Storey and R. Wang. Modeling quality requirements in conceptual database design. *Total Data Quality Management, Working Paper Series*: TDQM-02-94. Online , http: web.mit..edu tdqm www wp94.html 1997, July , 1994.

D. Strong, Y. Lee and R. Wang. Data quality in context. *Communications of the ACM*, 40(5), (1997), 103–110.

C. Tenopir. Database quality revisited. *Library Journal*, 10 01 90, pp. 64–67.

R. Wang. A product perspective on total data quality management. *Communications of the ACM*, 41(2), (1998), 58–65.

Y. Wand and R. Wang. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, November, 1996.

R. Wang, H. Kon and S. Madnick. Data quality requirements analysis and modeling. *9th International Conference on Data Engineering*, (1993), pp. 670–677.

R. Wang, D. Strong and L. Guarascio. Beyond accuracy: what data quality means to data consumers. *Journal of Management Information Systems*, 12(4), (1996), 5–34.

R. Wang, D. Strong, B. Kahn and Y. Lee. An information quality assessment methodology: extended abstract. *Proceedings of the 1999 Conference on Information Quality*, Yang Lee and Giri Tayi, eds., Cambridge, MA, 1999, pp. 258–263.

R. Wang, V. Storey and C. Firth. A framework for analysis of data quality research. *IEEE Transactions on Knowledge and Data Engineering*, 7(4), (1995), 349–372.

R. Zmud. Concepts, theories and techniques: an empirical investigation of the dimensionability of the concept of information. *Decision Design*, 9(2), (1978), 187–195.